A LIKELIHOOD BASED FRAMEWORK FOR DATA INTEGRATION WITH APPLICATION TO EQTL MAPPING

A Dissertation

by

SHUO FENG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jianhua Huang
Co-Chair of Committee,	Jianhua Hu
Committee Members,	Michael Sherman
	Guoyao Wu
Head of Department,	Simon Sheather

August 2014

Major Subject: Statistics

ABSTRACT

We develop a new way of thinking about and integrating gene expression data (continuous) and genomic information data (binary) by jointly compressing the two data sets and embedding their signals in low dimensional feature spaces with an information sharing mechanism, which connects the continuous data to the binary data, under the penalized log-likelihood framework. In particular, the continuous data are modeled by a Gaussian likelihood and the binary data are modeled by a Bernoulli likelihood which is formed by transforming the feature space of the genomic information with a logit link. The smoothly clipped absolute deviation (SCAD) penalty, is added on the basis vectors of the low dimensional feature spaces for both data sets, which is based on the assumption that only a small set of genetic variants are associated with a small fraction of gene expression and the fact that those basis vectors can be interpreted as weights assigned on the genetic variants and gene expression similar to the way the loading vectors of principal component analysis (PCA) or canonical correlation analysis (CCA) are interpreted. Algorithmically, a Majorization-Minimization (MM) algorithm with local linear approximation (LLA) to SCAD penalty is developed to effectively and efficiently solve the optimization problem involved, which produces closed-form updating rules. The effectiveness of our method is demonstrated by simulations in various setups with comparisons to some popular competing methods and an application to eQTL mapping with real data.

DEDICATION

To my wife Aonan, and my daughters, Angelica and Vivian

ACKNOWLEDGEMENTS

I would like to take this chance to express my gratitude to Dr. Jianhua Huang for all the well designed training opportunities from which I have gained not only knowledge but also the capabilities I can benefit from for time that lies before me. And many thanks go to Dr. Jianhua Hu for her support, constant input and guidance she has put into this project and the valuable insights into the potential application of our developed methodology to eQTL mapping. I am also very thankful for all the biological knowledge Dr. Guoyao Wu has given to me and all the intricate biochemical concepts he has explained to me with great patience. And I really appreciate all the statistical concepts and methodology Dr. Michael Sherman has shed light on in his wonderful and memorable courses. In addition, I wish to thank Dr. Lan Zhou for her effort in the algorithmic efficiency and advice on the design of simulation studies. Finally, thanks also to my friends and colleagues for their selfless help in work and in life.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

With advances in biomedical sciences and technologies, such as gene expression and single nucleotide polymorphism (SNP) microarray technology, collecting gene expression data with massive amount of measurements and high-density genotype data associated with the same set of individuals becomes commonplace, though analyzing such data by integrating the genomic and gene expression information could be challenging and involving. A genome-wide association study (GWA study or GWAS) has become a widely used way to examine a large set of common genetic variants in a set of individuals of interest to detect any genetic variant associated with a trait. In general, GWAS focuses on identifying associations between SNPs and gene expression levels of many traits. One popular approach to achieving that goal is expression quantitative trait loci (eQTL) mapping, which searches for significant associations between genetic variants and gene expression in hope of revealing the genetic factors causing certain diseases. And a major challenge of eQTL mapping approach results from the usually massive search space for potential eQTLs, by the nature of genomic or gene expression data.

A comparatively straightforward and commonly taken method for eQTL mapping is by the traditional approach where one enumerates all possible pairs of trait and genetic variant, looking for significant associations. One popular way of doing that is, for each SNP-transcript pair, a t-test is done on the transcript based on the grouping information in the SNP. Specifically, each trait is dichotomized based on the coding in the SNP and significance is claimed after some multiple comparison adjustment. And this can also be done by forming a simple linear regression model as $\mathbf{y}_i = \alpha + \beta \mathbf{x}_j + \epsilon$, where $\mathbf{y}_i$ is the ith transcript (gene expression) and $\mathbf{x}_j$ is the jth

SNP. An association is claimed if the null hypothesis $\beta = 0$ is rejected after multiple testing adjustments. The slope equal to zero can be tested by a t-test, F-test or a likelihood ratio test, each of which is a function of Pearson's correlation between $\mathbf{y}_i$ and $\mathbf{x}_j$. Shabalin (2012) developed the matrix eQTL for efficient calculation. The way to detect association by calculating pairwise correlations between the genotypes (genetic variants) and expression levels of genes is also demonstrated by Montgomery et al. (2010), Listgarten et al. (2010), and Chen et al. (2008), etc. Though traditional approaches are easy to implement, they have several disadvantages. For example, those methods assume genetic variants (loci) are independent, as a result, a single genetic variant explains only a small proportion of the variations in phenotypes which can be filtered out after multiple testing corrections. And they also overlook the correlation structure of gene expression. Moreover, simple linear regression is asymmetric as $\mathbf{y}_i$ is regressed on $\mathbf{x}_j$, but there is no clear reason of doing that rather than the other way around. Besides, multiple comparison adjustment is needed as independent testing for association for every transcript-SNP pair is done.

A more involving approach of eQTL mapping is by sparse multivariate regression which is able to take into account co-regulation effects of genetic loci and correlation structure of gene expression and likely increase the power to detect comparatively weak associations missed under independence assumption. Assume the matrix of genotypes (SNPs) has dimensionality $\mathbf{X}_{n\times J}$ and the matrix of gene expression is as $\mathbf{Y}_{n\times K}$. Consider the multiple-input-multiple-output (MIMO) linear system, $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B}_{J\times K} = \{\mathbf{b}^1, \cdots, \mathbf{b}^K\}$ with $\mathbf{b}^k$ defined as the kth column of $\mathbf{B}$ denoting the association coefficients of all genetic variants to that particular kth trait (Chen et al., 2008). For the estimation of the slope matrix, regularization is always incorporated in accordance with the sparsity assumption that only a small fraction of genetic variants are believed to be associated with differences in a subset

of gene expression. On the other hand, regularization is also needed with large K or J as OLS does not work when singularity emerges. Many methods under sparse multivariate regression framework have been proposed recently, improving over the defects of the traditional methods. Here we list a few, for example, Kim et al. (2009) developed the unweighted graph guided fused Lasso (G_uFLASSO) which searches for associations between a genetic variant and a subset of phenotypes rather than a single one. G_uFLASSO starts by constructing a phenotype correlation graph with nodes defined by the K traits and edges (in edge set E) representing the connectivity between the nodes, where node m and l are connected if the magnitude of the correlation of the mth and lth trait r_{ml} is above a certain threshold. Parameter estimation is achieved by solving:

$$\hat{\mathbf{B}} = minimize_{\mathbf{B}} \sum_{k=1}^{K} ||\mathbf{y}^k - \mathbf{X}\mathbf{b}^k||_2^2 + \lambda||\mathbf{B}||_1 + \gamma \sum_{(m,l)\in E} \sum_{j=1}^{J} |b_{jm} - sign(r_{ml})b_{jl}|, \quad (1.1)$$

in which Lasso penalty sets many of the coefficients to exact zero and the generalized fusion penalty brings closer the values of the coefficients of highly correlated gene expression for each genetic variant. However, this method exploits only the topology of the graph (presence or absence of edges) but the strength of connections. A direct generalization of G_uFLASSO is the weighted graph guided fused Lasso (G_wFLASSO), which takes into consideration the edge weights in addition to graph topology, by the same authors. Its parameter estimation is given by:

$$\hat{\mathbf{B}} = minimize_{\mathbf{B}} \sum_{k=1}^{K} ||\mathbf{y}^k - \mathbf{X}\mathbf{b}^k||_2^2 + \lambda||\mathbf{B}||_1 + \gamma \sum_{e_{m,l}\in E} w(e_{m,l}) \sum_{j=1}^{J} |b_{jm} - sign(r_{ml})b_{jl}|, \quad (1.2)$$

where $w(e_{m,l})$ is a weight, which can be simply defined as $|r_{ml}|$, assigned to the edge connecting node m and l, controlling the fusion effect. Chen et al. (2012) improved

further by the two-graph guided multi-task Lasso which is designed for the cases where a collection of genetic variants jointly regulate the co-expression of a set of genes (subnetwork to subnetwork) compared to only subnetwork to a single genetic marker structure previously. The slopes are estimated by:

$$minimize_{\mathbf{B}}||\mathbf{Y}-\mathbf{XB}||_F^2+\lambda||\mathbf{B}||_1+\gamma_1\sum_{e_{m,l}\in E_1}w(e_{m,l})\sum_{j=1}^{J}|b_{jm}-sign(r_{ml})b_{jl}| \\ +\gamma_2\sum_{e_{f,g}\in E_2}w(e_{f,g})\sum_{k=1}^{K}|b_{fk}-sign(r_{fg})b_{gk}| \quad (1.3)$$

From (1.3), we can see, for each genetic variant, the first generalized fusion penalty brings closer the values of the coefficients of highly correlated traits; for each trait, the second generalized fusion penalty smoothes over the values of the coefficients of highly correlated genetic variants, where r_{fg} is the Pearson's correlation between the fth and gth genetic marker as if they were continuous. Sparse multivariate regression approaches are more versatile than the traditional methods but they also have some limitations. For instance, as regression approaches, they are again not symmetrical and there is no clear reason why regression coefficients are only put on the genetic markers (SNPs). Moreover, they have scalability issues as calculating correlations for all possible pairs of traits and all pairs of genetic markers can be prohibitive for large data sets.

Canonical correlation analysis (CCA), proposed by Hotelling (1936), finds a linear combination of a set of variables measuring one set of individuals and a linear combination of another set of variables measuring the same set of individuals such that the two linear combinations of variables have the maximum correlation. Many extensions of CCA have been made, in particular, regularized CCA by introducing penalties, when it comes to analyzing data sets where the number of variables is

much larger than the number of observations (high-dimension low-sample-size). For instance, the regularized CCA (rCCA) by González et al. (2008) which is a direct extension to the regular CCA based on the definition; a penalized matrix decomposition (PMD) approach by Witten et al. (2009) which lies in the framework of convex optimization; the list continues with Parkhomenko et al. (2009), Lykou and Whittaker (2010), Waaijenborg et al. (2008), etc. Note that CCA deals with two sets of measurements on the same set of samples, having similarity to eQTL mapping setup, although not designed for that purpose, might be suitable for an eQTL task. And in reality, some proposed sparse canonical correlation analysis (SCCA) methods can be found in the literature in handling eQTL mapping tasks. One example is the group-structured SCCA for eQTL mapping by Chen et al. (2012), which assumes the prior structural knowledge on genes is available, e.g., biological pathways (a group of genes involved in a particular biological process). A combination of Lasso penalty, overlapping group Lasso penalty and ridge penalty is added on the loadings, encouraging sparsity and grouping effects. Similarly, Lin et al. (2013) solves CCA from the perspective of best rank one matrix approximation with balanced Lasso and group Lasso penalties which also assumes the structural knowledge on groups are known a priori. SCCA based approaches provide sequential procedures which extract one layer of information at a time. It is worth mentioning that SCCA approaches are symmetric, meaning they treat the two sets of measurements equally instead of using one set of measurements (genetic markers) to explain the other (gene expression). However, the two SCCA approaches mentioned still suffer from some limitations. For example, they treat the binary genetic markers as if they were continuous which incurs clumsiness when it comes to interpretation as it is hard to justify linear combinations of binaries. Apart from that, if group structures are not known in advance, the estimation part can be very computationally costly, if not impossible, as graphs

have to be formed by correlation information for $\mathbf{X}$ and $\mathbf{Y}$ respectively in order to apply generalized fused Lasso penalty.

Combining the features and principles of eQTL and CCA, we develop a new way of thinking about and integrating gene expression and genomic information with computational easiness and appealing interpretations. The idea is the following. Suppose there is a gene expression data set which consists of continuous entries and a genomic variants data set (SNPs) which is binary. The two data sets are jointly compressed by being embedded in low dimensional feature spaces with an information sharing mechanism, which connects the continuous data to the binary data, under the penalized log-likelihood framework. In particular, the continuous data are modeled by a Gaussian likelihood and the binary data are modeled by a Bernoulli likelihood which is formed by transforming the feature space of SNP genotypes with a logit link. The two log-likelihoods are carefully balanced, suggested by Rish et al. (2008) with a completely different objective, such that information from one data set does not dominate over the other. The smoothly clipped absolute deviation (SCAD) penalty, proposed by Fan and Li (2001), is added on the basis vectors of the low dimensional feature spaces for both data sets, which is based on the assumption that only a small set of genetic variants are associated with a small fraction of gene expression and the fact that those basis vectors can be interpreted as weights assigned on the genetic variants and gene expression similar to the way the loading vectors of principal component analysis (PCA) or CCA are interpreted. Apart from that, sparsity induced by penalty could also improve computational stability and efficiency. Algorithmically, the log-Bernoulli likelihood is not easy to differentiate, so a Majorization-Minimization (MM) algorithm (Hunter and Lange, 2004) is applied and the negative log-Bernoulli likelihood is hence bounded constantly by carefully defined quadratic majorizing surrogate functions. To ease the computation further,

local linear approximation (LLA) to SCAD penalty (Zou and Li, 2008) is employed and the penalty terms can also be majorized by quadratic surrogates. An iterative alternating algorithm for minimizing quadratic surrogates of the penalized negative joint log-likelihood is developed from the essence of the sparse logistic PCA algorithm by Lee et al. (2010). The developed algorithm only involves matrix operations, which produces neat closed-form updating steps, compared to those relatively hard-to-implement algorithms of most penalized eQTL mapping or sparse CCA. Further, after convergence of the algorithm, we come up with a way to order the importance of information contained in a layer-by-layer structure as what to expect in PCA or singular value decomposition (SVD), where a layer consists of a set of genetic variants (SNPs) and a set of gene expression. So genetic variants and gene expression associations are included in layers, which are of different importance and can overlap, and the importance of a layer is quantified as the variability explained jointly by the group of genetic variants and the group of gene expression in that particular layer. Concisely, our method is designed for the cases in which a set of genetic variants jointly associate with and co-regulate the expression of a set of genes and we can have a sequence of associations or co-regulation relationships ordered decreasingly in importance. In summary, our approach is accompanied by many merits. First, it is symmetric by treating two data sets with equal importance. Second, it works in the feature space of the SNP matrix instead of pretending it were continuous. Third, it is able to extract all layers of information simultaneously where each layer incorporates a group-to-group association structure and an order of importance can be established among layers. And overlapping structures are allowed among layers. Besides, our algorithm has closed-form updating procedures which involve only matrix operations computationally, so it is easy to program and efficient to execute. Finally, our method has loads of extensibility. For example, a different link function can be used for

modeling the binary data set and fusion penalties can also be applied to encourage smoothness in consecutive loading components for an improved grouping effect.

2. THE MODEL WITH ASSOCIATED PENALIZED LIKELIHOOD

2.1 The Model Setup and Its Penalized Likelihood

Suppose $\mathbf{X} = (X_{ij})$ is an $n \times d_1$ data matrix with continuous entries. Each row of $\mathbf{X}$ represents an observation and each column represents a particular measurement or variable. Each entry of $\mathbf{X}$ is an independent Gaussian random variable with individual mean and variance. We assume:

$$E(\mathbf{X}) = \mathbf{1_n} \otimes \mu^T + \mathbf{A}\mathbf{B}^T \tag{2.1}$$

and the ijth entry of $\mathbf{X}$, $x_{ij} = \mu_j + \mathbf{a}_i^T \mathbf{b}_j + \epsilon_{ij}$, where ϵ_{ij}'s are i.i.d. and $\epsilon_{ij} \sim \mathcal{N}(0,\, \sigma^2)$. Note that $\mathbf{a}_i^T$ is the ith row of $\mathbf{A}$ and $\mathbf{b}_j^T$ is the jth row of $\mathbf{B}$. In the equation above, μ is a $d_1 \times 1$ vector representing the overall mean of the expectation of $\mathbf{X}$, $\mathbf{A}$ is an $n \times k$ matrix and $\mathbf{B}$ is $d_1 \times k$, where k is the dimensionality of a low-dimensional subspace. From this setup, we can see it is exactly the model form of principal component analysis (PCA), as each column of $\mathbf{B}$ can be considered a loading vector and matrix $\mathbf{A}$ is the score matrix. However, the estimation procedure, as discussed later, is likelihood based. By normality assumption, the log-likelihood of $\mathbf{X}$, after suppressing the constant term, can be written as:

$$\begin{aligned} l_X(\mu,\, \mathbf{A},\, \mathbf{B},\, \sigma^2) &= log\left(\prod_{j=1}^{d_1} \prod_{i=1}^{n} (2\pi\sigma^2)^{-\frac{1}{2}} exp\left(\frac{-\left(x_{ij} - (\mu_j + \mathbf{a}_i^T \mathbf{b}_j)\right)^2}{2\sigma^2} \right) \right) \\ &= -\frac{nd_1}{2} log\sigma^2 - \frac{1}{2\sigma^2} \sum_{j=1}^{d_1} \sum_{i=1}^{n} \left(x_{ij} - (\mu_j + \mathbf{a}_i^T \mathbf{b}_j)\right)^2 + C. \end{aligned} \tag{2.2}$$

Further suppose there is another data set $\mathbf{Y} = (Y_{ij})$, an $n \times d_2$ binary matrix consisting of 0's and 1's. The second data set $\mathbf{Y}$ concerns the same set of observations

with a different set of measurements or variables which are binary, for example, the single-nucleotide polymorphisms (SNP). We assume the entries of $\mathbf{Y}$ are independent Bernoulli random variables with individual success probabilities, in particular, $Y_{ij} \sim Bernoulli(\pi_{ij})$. Consider the canonical parameter $\theta_{ij} = log\{\pi_{ij}/(1-\pi_{ij})\}$, which is the logit transformation of π_{ij}. Then π_{ij} can be expressed by $\pi_{ij} = \pi(\theta_{ij})$, where $\pi(x) = \{1 + exp(-x)\}^{-1}$. We have: $P(Y_{ij} = y_{ij}) = \pi(\theta_{ij})^{y_{ij}}(1-\pi(\theta_{ij}))^{1-y_{ij}} = \pi(q_{ij}\theta_{ij})$, where $q_{ij} = 2y_{ij} - 1$ (Lee et al., 2010). Similarly, we embed the features $\mathbf{\Theta}_{n\times d_2}$, instead of the data matrix $\mathbf{Y}$, into a low-dimensional subspace. Specifically,

$$\mathbf{\Theta} = \mathbf{1_n} \otimes \nu^T + \mathbf{AC}^T \tag{2.3}$$

In (2.3), ν is a $d_2 \times 1$ vector representing the overall mean of the features of $\mathbf{Y}$, $\mathbf{A}$ is the same as it is in (1) and $\mathbf{C}$ is a $d_2 \times k$ matrix which can be treated as the loadings in the feature space of $\mathbf{Y}$, inherited from logistic PCA (Lee et al., 2010). One thing worth mentioning is that the same $\mathbf{A}$ is used as a link connecting the information contained in $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{A}$, non-random, can be treated as latent quantities and interpreted as intrinsic attributes associated with the set of observations in the study. The log-likelihood of $\mathbf{Y}$ can be neatly written as:

$$l_Y(\nu,\, \mathbf{A},\, \mathbf{C}) = \sum_{j=1}^{d_2}\sum_{i=1}^{n} log\pi\big(q_{ij}(\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\big) \tag{2.4}$$

where we use the fact that a particular entry of $\mathbf{\Theta}$ is expressed as $\theta_{ij} = \nu_j + \mathbf{a}_i^T\mathbf{c}_j$. Here we notice that the above model is only estimable up to $\mathbf{AB}^T$ and $\mathbf{AC}^T$ which is discussed in detail in Section 2.4 and we also propose a way to tackle the identifiability issue in Section 3.

The overall log-likelihood of both $\mathbf{X}$ and $\mathbf{Y}$ is defined as:

$$l(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2) = \alpha l_X(\mu,\, \mathbf{A},\, \mathbf{B},\, \sigma^2) + l_Y(\nu,\, \mathbf{A},\, \mathbf{C}) \tag{2.5}$$

where α is a balancing parameter which makes the magnitude of l_X and that of l_Y comparable. The reason of balancing the log-likelihoods is to prevent one log-likelihood dominating over the other one during the optimization and estimation process to be discussed in Section 3, as X_{ij}'s are normals and Y_{ij}'s are Bernoulli and their density/mass functions have different ranges. We consider $\mathbf{X}$ and $\mathbf{Y}$ of equal importance, for our purpose is to investigate the connection between the two data sets and the information shared. From our empirical studies, a not so carefully chosen α could incur total loss of information in one of the two data sets. A similar treatment can be found in Rish et al. (2008) with a different objective. The principle of α estimation is discussed in the next section and the technical details are disclosed in Section 3.

For better interpretability and computational efficiency, it is desirable to regularize the overall log-likelihood by adding penalties to get sparse loading matrices $\mathbf{B}$ and $\mathbf{C}$. Holding $\mathbf{a}_i$'s fixed as if they were observable, the model analogy to regression also suggests use of penalty on loadings. Penalty functions of $\mathbf{B}$ and $\mathbf{C}$ are denoted by $P_\lambda(\mathbf{B})$ and $P_\gamma(\mathbf{C})$, respectively, where λ and γ are vectors of tuning parameters of length k, hence, in principle, each column of $\mathbf{B}$ or $\mathbf{C}$ is allowed to have a different amount of penalization for maximal flexibility. In this paper, we use the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001), which shrinks small values to exact zero and is asymptotically unbiased for large values, compared to LASSO (Tibshirani, 1996), one popular alternative having biasedness

for large values. Then it comes to the following criterion function:

$$S(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2) = -l(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2) + nP_\lambda(\mathbf{B}) + nP_\gamma(\mathbf{C}) \tag{2.6}$$

In (2.6), the negative log-likelihood can be perceived as a loss function and the two penalty terms compete with the loss function. The target is to minimize the penalized negative log-likelihood $S(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2)$. The detailed form of the penalty functions and the way to minimize the criterion function are discussed in Section 3.

2.2 Estimation of α

As mentioned above, estimating α accurately plays an important role in estimating all the parameters of interest. In this subsection, we discuss the principle of α estimation briefly and the technical details and algorithms are left for further discussion in Section 3. The idea is as the following. The first step is to extract the main signals in $\mathbf{X}$ and $\mathbf{Y}$ separately. In order to achieve that, we compress $\mathbf{X}$ with penalty to itself and do the some thing to $\mathbf{Y}$. Define, for $\mathbf{X}$ alone:

$$\begin{aligned} l_X(\mu_X,\, \mathbf{A}_X,\, \mathbf{B}_X,\, \sigma_X^2) = -\frac{nd_1}{2} log\sigma_X^2 - \frac{1}{2\sigma_X^2}\sum_{j=1}^{d_1}\sum_{i=1}^{n}\left(x_{ij} - (\mu_{Xj} + \mathbf{a}_{Xi}^T\mathbf{b}_{Xj})\right)^2 \\ - \frac{nd_1}{2} log2\pi \end{aligned} \tag{2.7}$$

which is almost identical to (2.2) with the constant figured out exactly and subscript "X" indicating all the parameters are estimated by $\mathbf{X}$ alone. All the parameters have the same dimensions as they do in (2.2), especially, the same dimensionality k is used. Consider using L_1 penalty to get a sparse loading $\mathbf{B}_X$ for individual data compression purpose, as in the LASSO regression, we minimize the following

criterion function:

$$S_X(\mu_X,\, \mathbf{A}_X,\, \mathbf{B}_X,\, \sigma_X^2) = -l_X(\mu_X,\, \mathbf{A}_X,\, \mathbf{B}_X,\, \sigma_X^2) + nP_{X\lambda}(\mathbf{B}_X) \tag{2.8}$$

where

$$P_{X\lambda}(\mathbf{B}_X) = \sum_{l=1}^{k} \lambda_l ||\tilde{\mathbf{b}}_{Xl}||_1 \tag{2.9}$$

and $\tilde{\mathbf{b}}_{Xl}$ is the lth column of $\mathbf{B}_X$. Similarly define:

$$l_Y(\nu_Y,\, \mathbf{A}_Y,\, \mathbf{C}_Y) = \sum_{j=1}^{d_2} \sum_{i=1}^{n} log\pi\big(q_{ij}(\nu_{Y\,j} + \mathbf{a}_{Y\,i}{}^T \mathbf{c}_{Y\,j})\big) \tag{2.10}$$

We use the sparse logistic PCA setup (Lee et al., 2010) and the algorithm proposed in that referenced paper (Algorithm 5) for the compression of $\mathbf{Y}$, which is achieved by minimizing the criterion function below:

$$S_Y(\nu_Y,\, \mathbf{A}_Y,\, \mathbf{C}_Y) = -l_Y(\nu_Y,\, \mathbf{A}_Y,\, \mathbf{C}_Y) + nP_{Y\gamma}(\mathbf{C}_Y) \tag{2.11}$$

where

$$P_{Y\gamma}(\mathbf{C}_Y) = \sum_{l=1}^{k} \gamma_l ||\tilde{\mathbf{c}}_{Yl}||_1 \tag{2.12}$$

and $\tilde{\mathbf{c}}_{Yl}$ is the lth column of $\mathbf{C}_Y$. Again, all the parameters are of the same dimensions as in (2.4). The second step is solving the optimization problems in (2.8) and (2.11) with optimally chosen tuning parameters (see more details in Section 3) and, with all the relevant parameters estimated, calculate $l_X(\mu_X,\, \mathbf{A}_X,\, \mathbf{B}_X,\, \sigma_X^2)$ and, $l_Y(\nu_Y,\, \mathbf{A}_Y,\, \mathbf{C}_Y)$ as defined in (2.7) and (2.10). Then α is calculated as the absolute value of the ratio of the two log-likelihoods, specifically, $\alpha = |l_Y/l_X|$, which can be interpreted as the relative magnitude of two log-likelihoods based on main signals

from $\mathbf{X}$ and $\mathbf{Y}$.

2.3 Choosing the Penalty Parameters and Determining the Dimensionality k

For the optimization problem in (2.6), we have to provide the two penalty vectors λ and γ as input before estimation procedures commence. As we mentioned, we can use different penalty parameters on different columns of $\mathbf{B}$ and $\mathbf{C}$ for maximal flexibility of the methodology, but from here on we only consider using a single penalty parameter λ on all loadings in $\mathbf{B}$ and a single γ on all loadings in $\mathbf{C}$. This simplification will in general reduce the computational burden substantially. From the property of SCAD, a larger value of λ and/or γ reduces the model complexity by setting more entries in $\mathbf{B}$ and/or $\mathbf{C}$ to zeros, but a less good fit of the model is a price to pay for enjoying a simpler model. These two competing factors are compromised, for fixed k, by minimizing the following fractional BIC criterion, for different combinations of λ and γ, which is a generalization of the BIC criterion proposed by Lee et al. (2010):

$$BIC(\lambda, \gamma) = -2l(\mu, \nu, \mathbf{A}, \mathbf{B}, \mathbf{C}, \sigma^2) + f_d \times log\, n \times m(\lambda, \gamma), \tag{2.13}$$

where f_d is a fractional number that $0 < f_d \leq 1$ and $m(\lambda, \gamma)$ is a measure of the degrees of freedom which is defined as $m(\lambda, \gamma) = d_1 + d_2 + nk + |\mathfrak{B}(\lambda)| + |\mathfrak{C}(\gamma)|$, where d_1 is the length of vector μ, d_2 is the length of vector ν, nk is the total number of elements in A, $|\mathfrak{B}(\lambda)|$ is the cardinality of the index set $\mathfrak{B}(\lambda)$ of the nonzero entries in $\mathbf{B}$ when the penalty parameter is λ, and $|\mathfrak{C}(\gamma)|$ is the cardinality of the index set $\mathfrak{C}(\gamma)$ of the nonzero entries in $\mathbf{C}$ when the penalty parameter is γ. The reason of introducing f_d to discount the latter part of the criterion function is that the BIC criterion proposed by Lee et al. (2010) tends to select a tuning parameter pair that over-penalizes the overall log-likelihood, resulting in loading matrices of

exact zeros which makes the model degenerate. The fractional number f_d should be pre-specified before choosing the tuning parameters and we propose that f_d is selected as the largest value possible before the model becomes degenerate. With f_d fixed, we propose an alternating searching scheme for the optimal pair of the tuning parameters λ and γ, as a two-dimensional grid search can be very computationally intensive. Based on our empirical studies on both simulated and real data sets, results from the alternating search coincide with the optima from the two-dimensional grid search for most of the time and, if not, the discrepancy measured by (2.13) is very small. As for the determination of k, we can just fix it at a moderate value, say, $k = 10$ or 20. The searching procedure for the tuning parameter pair is listed below:

1. Fix k at a moderate value, say, $k = 10$ or 20.

2. Set a value to f_d, e.g., $f_d = 0.001, 0.005, 0.01, 0.05, 0.1, \cdots, 1$.

3. For a fixed f_d, search for the optimal value of each tuning parameter over a collection of candidate values alternately. For instance, set $\lambda = 10^{-7}$, search over all γ values from $10^{-7}, 10^{-6}, \cdots, 0.1, 1$ and the optimal γ is the one that minimizes the fractional BIC. Once the optimal γ is decided, fix γ at that value and repeat the same procedure for optimal λ. Alternate searching for λ and γ until the change of fractional BIC is smaller than a pre-specified threshold. Record the optimal pair and the corresponding f_d.

4. Repeat step 3 for different f_d.

5. The optimal pair of λ and γ is the one corresponding to the largest f_d such that the model is non-degenerate.

2.4 Model Identifiability

For the multiplicative nature of the model, the effects of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ can not be easily separated from $\mathbf{AB}^T$ or $\mathbf{AC}^T$. In order to understand the situation we

face with identifiability issue, we need to shed some light on how the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are estimated, with details discussed in Section 3. Specifically, matrix $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are estimated by an alternating procedure where one is estimated in turn with the other two kept fixed. After convergence, each column of the estimated $\mathbf{B}$ and $\mathbf{C}$ is re-scaled to have length unity, and then matrix $\mathbf{A}$ is re-estimated with the re-scaled $\mathbf{B}$ and $\mathbf{C}$ fixed. The reason we re-scale $\mathbf{B}$ and $\mathbf{C}$ is that we want to make them unit basis vectors, assigning weights to the variables continuous or binary, as if they were like the loading vectors of PCA, which is suggested by the analogy of our model to PCA in both setup and interpretation. However, it is worth pointing out that the re-scaled estimated $\mathbf{B}$ and $\mathbf{C}$ are not orthogonal, thus, the columns of the re-scaled estimated $\mathbf{B}$ and $\mathbf{C}$ constitute a set of basis vectors non-orthogonal for the features of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Now suppose we have $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ at convergence where matrix $\mathbf{B}$ and $\mathbf{C}$ have column length unity. It is clear that inserting, between $\mathbf{A}$ and $\mathbf{B}^T$, a product of the transpose of a matrix with orthogonal columns and that matrix or an invertible matrix and its inversion does not change the product of $\mathbf{AB}^T$, similarly, for $\mathbf{AC}^T$. For the case of inserting matrices with orthogonal columns, consider:

$$\begin{aligned}
\mathbf{AB}^T &= \mathbf{AM}^T\mathbf{D}_M\mathbf{D}_M^{-1}\mathbf{MB}^T \\
&= \mathbf{AM}^T\mathbf{D}_M(\mathbf{BM}^T\mathbf{D}_M^{-1})^T \\
&= (\mathbf{AM}^{-1}\mathbf{D}_M)(\mathbf{BM}^{-1}\mathbf{D}_M^{-1})^T \\
&= \mathbf{A}'\mathbf{B}'^T
\end{aligned} \tag{2.14}$$

In (2.14), $\mathbf{M}$ is a matrix with orthogonal columns and $\mathbf{D}_M^{-1}$, a diagonal matrix, is used to re-scale (make column length unity of) the transformed $\mathbf{B}$, $\mathbf{BM}^T$, which can be seen more clearly in the second equation. Notice that it is required that the

dimensionality of $\mathbf{B}\mathbf{M}^{-1}\mathbf{D}_M^{-1}$ should be the same as that of $\mathbf{B}$ as the former one is considered a new solution of $\mathbf{B}$ to the fixed quantity $\mathbf{A}\mathbf{B}^T$. And that requirement implicitly implies the matrix $\mathbf{M}$ must be a square matrix of size $k \times k$ and it is invertible and hence we have the third equation. We name the new solutions to $\mathbf{A}\mathbf{B}^T$ as $\mathbf{A}'$ and $\mathbf{B}'$ shown in the fourth equation of (14). Similarly, a matrix $\mathbf{Q}$ with orthogonal columns and a re-scaling diagonal matrix $\mathbf{D}_Q^{-1}$ can be found such that:

$$\begin{aligned}
\mathbf{A}\mathbf{C}^T &= \mathbf{A}\mathbf{Q}^T\mathbf{D}_Q\mathbf{D}_Q^{-1}\mathbf{Q}\mathbf{C}^T \\
&= \mathbf{A}\mathbf{Q}^T\mathbf{D}_Q(\mathbf{C}\mathbf{Q}^T\mathbf{D}_Q^{-1})^T \\
&= (\mathbf{A}\mathbf{Q}^{-1}\mathbf{D}_Q)(\mathbf{C}\mathbf{Q}^{-1}\mathbf{D}_Q^{-1})^T \\
&= \mathbf{A}'\mathbf{C}'^T
\end{aligned} \tag{2.15}$$

We need to pay attention to the commonality of $\mathbf{A}'$ in (2.14) and (2.15) since that deals with information sharing in our model, and this implies the equality of $\mathbf{A}\mathbf{M}^{-1}\mathbf{D}_M$ and $\mathbf{A}\mathbf{Q}^{-1}\mathbf{D}_Q$, which, in turn, implies $\mathbf{M} = \mathbf{D}_M\mathbf{D}_Q^{-1}\mathbf{Q}$ given $\mathbf{A}$ is of full column rank, which means each column of $\mathbf{M}$ is in proportion to the corresponding column of $\mathbf{Q}$ with possibly different proportions if $\mathbf{M}$ and $\mathbf{Q}$ exist. On the other hand, the argument is almost identical for inserting invertible matrices between $\mathbf{A}$ and $\mathbf{B}^T$ or $\mathbf{A}$ and $\mathbf{C}^T$. In the following, we present a theorem, saying that the signals will not be changed even with identifiability issues. Moreover, from a practical perspective, we do not need to worry too much about model identifiability as signals can always be captured, confirmed by our intensive empirical studies with various setups.

Theorem 2.4.1 *For a loading matrix $\mathbf{B}_{d\times k}$, define $\mathbf{B}' = \mathbf{B}\mathbf{N}$, where $\mathbf{N}$ is a $k \times k$ invertible matrix. The entries of $\mathbf{N}$ are i.i.d. generated from a distribution F. Then*

$\mathbf{B}$ *and* $\mathbf{B}'$ *have the same set of signals almost surely.*

Proof. First we prove the set of signals in $\mathbf{B}'$ is a subset of the set of signals in $\mathbf{B}$. We define that there is a signal at the ith variable in the loading matrix $\mathbf{B}$ if and only if there exists a j, such that the ijth entry of $\mathbf{B}$, b_{ij}, is not zero. Assume there is a signal at the ith variable in $\mathbf{B}'$, i.e., there exists a j such that $b'_{ij} \neq 0$. Denote the ith row of $\mathbf{B}$ by $\mathbf{B}_{i\cdot}$ and the jth column of $\mathbf{N}$ as $\mathbf{N}_{\cdot j}$. Thus, $\mathbf{B}_{i\cdot}\mathbf{N}_{\cdot j} = b'_{ij} \neq 0$, which implies $\mathbf{B}_{i\cdot} \neq 0$, which, in turn, implies an existing $b_{il} \neq 0$ for some l. And this indicates there is a signal in the ith variable in $\mathbf{B}$. Therefore we have proved the set of signals in $\mathbf{B}'$ is a subset of the set of signals in $\mathbf{B}$ and, moreover, we note that $P(\mathbf{B}_{i\cdot}\mathbf{N}_{\cdot j} \neq 0) = 1 - P(\mathbf{B}_{i\cdot}\mathbf{N}_{\cdot j} = 0) = 1$. Then, for the other direction, note that $\mathbf{B} = \mathbf{B}'\mathbf{N}^{-1}$ and apply the above argument procedure to prove the set of signals in $\mathbf{B}$ is a subset of the set of signals in $\mathbf{B}'$. Hence, we have the claimed results in the theorem.

If we replace the matrix $\mathbf{N}$ in the theorem by $\mathbf{M}^{-1}\mathbf{D}_M^{-1}$ or $\mathbf{Q}^{-1}\mathbf{D}_Q^{-1}$, we claim model identifiability issue does not change the set of signals contained in the loading matrices $\mathbf{B}$ or $\mathbf{C}$.

3. COMPUTATIONAL ALGORITHMS

In this section, we discuss about and disclose all the computational details for the procedures described in the previous section. Section 3.1 deals with how to optimize the criterion function in (2.6) and how to post-process the estimated matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ after convergence so that they can have natural interpretations of being scores or loadings, especially, how we can come up with a way to order the importance of information embedded in a layer-by-layer structure as what to expect in principal component analysis (PCA) or singular value decomposition (SVD). The know-how on the estimation of α mentioned in Section 2.2 is explained in Section 3.2 and the way of initialization to kick off the main algorithm is shown in Section 3.3.

3.1 The Main Algorithm

The direct minimization of the criterion function (2.6) can be hard and it does not seem to have an analytical solution even we put aside the penalty terms temporarily, since the primary challenge comes from differentiating the log-likelihood of the binary data matrix and the non-differentiability of the penalty functions to be discussed. Instead, we develop a Majorization-Minimization (MM) algorithm for that optimization problem, particularly, we carefully select quadratic auxiliary functions, which constantly act as upper bound of the log-likelihood of $\mathbf{Y}$, and we use another set of quadratic functions to bound the penalty terms from above. As a result, solving the minimization problem defined in (2.6) can be achieved by iteratively minimizing a bounding quadratic surrogate function to be defined in detail in this section, as differentiating a quadratic function is comparatively straightforward and closed-form solutions can be expected.

We begin with defining majorization functions. A function $h(x, y)$ is said to

majorize a function $f(x)$ at y if $h(y,y) = f(y)$ and $h(x,y) \geq f(x)$ for all x. Geometrically, for a fixed value of y, the function $h(x,y)$ with respect to x lies above the curve of $f(x)$ and is tangent to it when x assumes the same value as y. Suppose $f(x)$ is to be minimized. Starting with an initial value $x^{(0)}$ of $x = \underset{x}{argmin}\, f(x)$, $x^{(m+1)}$ is produced iteratively as $x^{(m+1)} = \underset{x}{argmin}\, h(x, x^{(m)})$, where $x^{(m+1)}$ is the estimate of x at the $(m+1)$th iteration given the value of the mth iteration $x^{(m)}$. It is not hard to see $f(x)$ is non-increasing under this iterative updating process, since $f(x^{(m+1)}) \leq h(x^{(m+1)}, x^{(m)}) \leq h(x^{(m)}, x^{(m)}) = f(x^{(m)})$, and thus the objective function $f(x)$ is guaranteed to converge to a local minimum under that iterative updating process described.

In order to find a proper majorizing function to ease the optimization of (2.6), we treat the log-likelihood terms and the penalty terms separately. Notice that the log-likelihood of $\mathbf{X}$ does not need any extra care to deal with since it is already in a quadratic form by nature. As contrast, for the log-likelihood of $\mathbf{Y}$, we consider the same majorizing function for the negative log inverse logit function, $-log\pi(\cdot)$, which is the key building block of the log-likelihood of $\mathbf{Y}$, as the one used by Lee et al. (2010) and De Leeuw (2006). Specifically, for a given y:

$$-log\pi(x) \leq -log\pi(y) - (1-\pi(y))(x-y) + \frac{1}{8}(x-y)^2, \tag{3.1}$$

and the equality holds when $x = y$. Completing the square for the right hand side of the above inequality leads to the following:

$$-log\pi(x) \leq -log\pi(y) + \frac{1}{8}\Big(x - y - 4\big(1-\pi(y)\big)\Big)^2 \tag{3.2}$$

By letting $x = q_{ij}\theta_{ij}$, $y = q_{ij}\theta_{ij}^{(m)}$, where $\theta_{ij}^{(m)}$ denotes the mth iteration's estimate of

θ_{ij}, and the fact that $q_{ij}^2 = 1$, we have, from (3.2):

$$-log\pi(q_{ij}\theta_{ij}) \leq -log\pi(q_{ij}\theta_{ij}^{(m)}) + \frac{1}{8}(\theta_{ij} - z_{ij}^{(m)})^2, \tag{3.3}$$

where $z_{ij}^{(m)} = \theta_{ij}^{(m)} + 4q_{ij}\big(1 - \pi(q_{ij}\theta_{ij}^{(m)})\big)$. After suppressing the expression of the constant terms irrelevant to estimating parameters of concern in the $(m+1)$th iteration, the negative log-likelihood of $\mathbf{Y}$, recalling $\theta_{ij} = \nu_j + \mathbf{a}_i^T\mathbf{c}_j$, is bounded as:

$$\begin{aligned} -l_Y(\nu,\, \mathbf{A},\, \mathbf{C}) &= -\sum_{j=1}^{d_2}\sum_{i=1}^{n} log\pi(q_{ij}\theta_{ij}) \\ &\leq \sum_{j=1}^{d_2}\sum_{i=1}^{n} \frac{1}{8}\big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\big)^2 + Cons. \end{aligned} \tag{3.4}$$

by the inequality in (3.3). As for the penalty terms, on the other hand, the SCAD penalty is employed on its own merits as we mentioned in Section 2. The SCAD penalty on a scalar β (e.g., a slope in regression) for a particular tuning parameter κ can be written as $p_\kappa(|\beta|)$, which is a concave function defined by $p_\kappa(0) = 0$ and for $|\beta| > 0$,

$$p'_\kappa(|\beta|) = \kappa I(|\beta| \leq \kappa) + \frac{(a\kappa - |\beta|)_+}{a-1} I(|\beta| > \kappa) \tag{3.5}$$

for some $a > 2$. Empirically, a is often set to 3.7 (Fan and Li, 2001), and we use $a = 3.7$ for all the calculations in this paper. The notation z_+ above denotes the positive part of z: $z_+ = z$ if $z > 0$ and $z_+ = 0$ otherwise. In order to fit in the rationale of our iterative estimation process, we consider local linear approximation (LLA) to SCAD penalty (Zou and Li, 2008). As a preliminary for the following argument, note the inequality:

$$|x| \leq \frac{x^2 + y^2}{2|y|}, \qquad y \neq 0, \tag{3.6}$$

gives an upper bound for $|x|$. Given a penalty parameter λ, consider the SCAD penalty for an arbitrary entry b_{jl} of the loading matrix $\mathbf{B}$, we have:

$$\begin{aligned} P_\lambda(|b_{jl}|) &\approx P_\lambda(|b_{jl}^{(m)}|) + P'_\lambda(|b_{jl}^{(m)}|)(|b_{jl}| - |b_{jl}^{(m)}|) \qquad\qquad b_{jl} \approx b_{jl}^{(m)} \\ &= P_\lambda(|b_{jl}^{(m)}|) - P'_\lambda(|b_{jl}^{(m)}|)|b_{jl}^{(m)}| + P'_\lambda(|b_{jl}^{(m)}|)|b_{jl}| \\ &\leq P'_\lambda(|b_{jl}^{(m)}|)\frac{b_{jl}^2 + {b_{jl}^{(m)}}^2}{2|b_{jl}^{(m)}|} + Cons. \\ &= \frac{P'_\lambda(|b_{jl}^{(m)}|)b_{jl}^2}{2|b_{jl}^{(m)}|} + Cons.2, \end{aligned} \tag{3.7}$$

where, in the first line, the penalty term is approximated by a local linear approximation at the point $|b_{jl}^{(m)}|$, the absolute value of the estimate of the jlth entry of $\mathbf{B}$ by the mth iteration. The fact of inequality (3.6) leads to the third line above, omitting the expression of irrelevant constants, and the fourth line only shows the terms containing b_{jl}. Hence, we have:

$$\begin{aligned} P_\lambda(\mathbf{B}) &= \sum_{j=1}^{d_1}\sum_{l=1}^{k} P_\lambda(|b_{jl}|) \\ &\leq \sum_{j=1}^{d_1}\sum_{l=1}^{k} \frac{P'_\lambda(|b_{jl}^{(m)}|)b_{jl}^2}{2|b_{jl}^{(m)}|} + Cons. \\ &= \sum_{j=1}^{d_1} \mathbf{b}_j^T \mathbf{D}_{1,j}^{(m)} \mathbf{b}_j + Cons., \end{aligned} \tag{3.8}$$

where $\mathbf{D}_{1,j}^{(m)} = diag\Big(\frac{P'_\lambda(|b_{j1}^{(m)}|)}{2|b_{j1}^{(m)}|}, \cdots, \frac{P'_\lambda(|b_{jk}^{(m)}|)}{2|b_{jk}^{(m)}|}\Big)$, a diagonal matrix. Similarly, the penalty term on $\mathbf{C}$ is bounded as:

$$P_\gamma(\mathbf{C}) \leq \sum_{j=1}^{d_2} \mathbf{c}_j^T \mathbf{D}_{2,j}^{(m)} \mathbf{c}_j + Cons., \tag{3.9}$$

where $\mathbf{D}_{2,j}^{(m)}$ is a diagonal matrix defined as $\mathbf{D}_{2,j}^{(m)} = diag\Big(\frac{P'_\gamma(|c_{j1}^{(m)}|)}{2|c_{j1}^{(m)}|}, \cdots, \frac{P'_\gamma(|c_{jk}^{(m)}|)}{2|c_{jk}^{(m)}|}\Big)$. Then, based on (3.4), (3.8) and (3.9), the criterion function S defined in (2.6) is bounded by a properly defined majorizing function as the following:

$$\begin{aligned} S(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2) \leq \frac{nd_1\alpha}{2} log\sigma^2 + \frac{\alpha}{2\sigma^2}\sum_{j=1}^{d_1}\sum_{i=1}^{n}\big(x_{ij} - (\mu_j + \mathbf{a}_i^T\mathbf{b}_j)\big)^2 \\ + \sum_{j=1}^{d_2}\sum_{i=1}^{n}\frac{1}{8}\big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\big)^2 + n\sum_{j=1}^{d_1}\mathbf{b}_j^T\mathbf{D}_{1,j}^{(m)}\mathbf{b}_j \\ + n\sum_{j=1}^{d_2}\mathbf{c}_j^T\mathbf{D}_{2,j}^{(m)}\mathbf{c}_j + Cons. \end{aligned} \tag{3.10}$$

We denote the right hand side of (3.10), which is an upper bound of S, by $g(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2|\mu^{(m)},\, \nu^{(m)},\, \mathbf{A}^{(m)},\, \mathbf{B}^{(m)},\, \mathbf{C}^{(m)},\, \sigma^{2(m)})$ and it is quadratic in each of μ, ν, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ when the other four are kept fixed. An alternating minimization procedure of g with respect to all the parameters of interest is derived below and we drop the superscript (m) whenever ambiguity does not rise. Set $x_{ij}^\dagger = x_{ij} - \mathbf{a}_i^T\mathbf{b}_j$, the optimal jth component of μ, $\hat{\mu}_j$ is obtained by:

$$\hat{\mu}_j = \underset{\mu_j}{argmin}\sum_{i=1}^{n}(x_{ij}^\dagger - \mu_j)^2 = \frac{1}{n}\sum_{i=1}^{n}x_{ij}^\dagger, \quad j = 1, \cdots, d_1, \tag{3.11}$$

and compactly the optimal $\hat{\mu}$ is obtained as $\hat{\mu} = \frac{1}{n}\mathbf{X}^{\dagger T}\mathbf{1}_n$ which is the column means of $\mathbf{X}^\dagger$. Define $z_{ij}^\dagger = z_{ij} - \mathbf{a}_i^T\mathbf{c}_j$, the optimal jth component of ν, $\hat{\nu}_j$ is given as:

$$\hat{\nu}_j = \underset{\nu_j}{argmin}\sum_{i=1}^{n}(z_{ij}^\dagger - \nu_j)^2 = \frac{1}{n}\sum_{i=1}^{n}z_{ij}^\dagger, \quad j = 1, \cdots, d_2, \tag{3.12}$$

and the optimal vector $\hat{\nu}$ is calculated as the column means of $\mathbf{Z}^\dagger$, that is, $\hat{\nu} =$

$\frac{1}{n}\mathbf{Z}^{\dagger T}\mathbf{1}_n$. The optimal $\hat{\sigma}^2$ is calculated straightforwardly as:

$$\begin{aligned}\hat{\sigma}^2 &= \underset{\sigma^2}{argmin}\Big\{nd_1 log\sigma^2 + \frac{1}{\sigma^2}\sum_{j=1}^{d_1}\sum_{i=1}^{n}(x_{ij}^{\dagger} - \mu_j)^2\Big\} \\ &= \frac{1}{nd_1}tr(\mathbf{M}\mathbf{M}^T) \end{aligned} \tag{3.13}$$

where $\mathbf{M} = \mathbf{X}^{\dagger} - \mathbf{1}_n \otimes \mu^T$. Calculating the optimal $\hat{\mathbf{A}}$ is more involving. First consider the optimal $\hat{\mathbf{a}}_i$, with $x_{ij}^* = x_{ij} - \mu_j$, $z_{ij}^* = z_{ij} - \nu_j$, the ith row of $\mathbf{X}^*$ denoted as $\mathbf{x}_i^{*T}$ and the ith row of $\mathbf{Z}^*$ denoted by $\mathbf{z}_i^{*T}$:

$$\begin{aligned}\hat{\mathbf{a}}_i &= \underset{\mathbf{a}_i}{argmin}\Big\{\frac{\alpha}{2\sigma^2}\sum_{j=1}^{d_1}(x_{ij}^* - \mathbf{a}_i^T\mathbf{b}_j)^2 + \frac{1}{8}\sum_{j=1}^{d_2}(z_{ij}^* - \mathbf{a}_i^T\mathbf{c}_j)^2\Big\} \\ &= \underset{\mathbf{a}_i}{argmin}\Big\{\frac{\alpha}{2\sigma^2}(\mathbf{x}_i^* - \mathbf{B}\mathbf{a}_i)^T(\mathbf{x}_i^* - \mathbf{B}\mathbf{a}_i) + \frac{1}{8}(\mathbf{z}_i^* - \mathbf{C}\mathbf{a}_i)^T(\mathbf{z}_i^* - \mathbf{C}\mathbf{a}_i)\Big\} \\ &= \left(\frac{\alpha}{\sigma^2}\mathbf{B}^T\mathbf{B} + \frac{1}{4}\mathbf{C}^T\mathbf{C}\right)^{-1}\left(\frac{\alpha}{\sigma^2}\mathbf{B}^T\mathbf{x}_i^* + \frac{1}{4}\mathbf{C}^T\mathbf{z}_i^*\right)\end{aligned} \tag{3.14}$$

Given $\hat{\mathbf{a}}_i$, we have $\hat{\mathbf{A}} = \big(\frac{\alpha}{\sigma^2}\mathbf{X}^*\mathbf{B} + \frac{1}{4}\mathbf{Z}^*\mathbf{C}\big)\big(\frac{\alpha}{\sigma^2}\mathbf{B}^T\mathbf{B} + \frac{1}{4}\mathbf{C}^T\mathbf{C}\big)^{-1}$, then $\hat{\mathbf{A}}$ is orthonormalized by Gram-Schmidt process in order to ease the estimation of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ as discussed in the following. Starting with denoting the jth column of $\mathbf{X}^*$ as $\tilde{\mathbf{x}}_j^*$ and utilizing the fact that $\mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}$, this leads to:

$$\begin{aligned}\hat{\mathbf{b}}_j &= \underset{\mathbf{b}_j}{argmin}\Big\{\frac{\alpha}{2\sigma^2}\sum_{i=1}^{n}(x_{ij}^* - \mathbf{a}_i^T\mathbf{b}_j)^2 + n\mathbf{b}_j^T\mathbf{D}_{1,j}^{(m)}\mathbf{b}_j\Big\} \\ &= \underset{\mathbf{b}_j}{argmin}\Big\{\frac{\alpha}{2\sigma^2}\big(\tilde{\mathbf{x}}_j^* - \mathbf{A}\mathbf{b}_j\big)^T\big(\tilde{\mathbf{x}}_j^* - \mathbf{A}\mathbf{b}_j\big) + n\mathbf{b}_j^T\mathbf{D}_{1,j}^{(m)}\mathbf{b}_j\Big\} \\ &= \alpha(\alpha\mathbf{I} + 2n\sigma^2\mathbf{D}_{1,j}^{(m)})^{-1}\mathbf{A}^T\tilde{\mathbf{x}}_j^*\end{aligned} \tag{3.15}$$

Noticing the matrix to be inverted is a diagonal matrix by the definition of $\mathbf{D}_{1,j}^{(m)}$, we

can figure out the expression of $\hat{\mathbf{b}}_j$ component-wise:

$$\hat{b}_{jl} = \frac{\alpha|b_{jl}^{(m)}|}{n\sigma^2 P'_\lambda(|b_{jl}^{(m)}|) + \alpha|b_{jl}^{(m)}|} g_{jl}, \quad j = 1, \cdots, d_1, \quad l = 1, \cdots, k, \tag{3.16}$$

where g_{jl} is the jlth entry of the matrix $\mathbf{G} = \mathbf{X}^{*T}\mathbf{A}$. Analogously, define the jth column of $\mathbf{Z}^*$ as $\tilde{\mathbf{z}}_j^*$, then minimization with respect to $\mathbf{c}_j$ produces:

$$\begin{aligned}
\hat{\mathbf{c}}_j &= \underset{\mathbf{c}_j}{argmin}\Big\{ \sum_{i=1}^{n} \frac{1}{8}(z_{ij}^* - \mathbf{a}_i^T\mathbf{c}_j)^2 + n\mathbf{c}_j^T\mathbf{D}_{2,j}^{(m)}\mathbf{c}_j \Big\} \\
&= \underset{\mathbf{c}_j}{argmin}\Big\{ \frac{1}{8}(\tilde{\mathbf{z}}_j^* - \mathbf{A}\mathbf{c}_j)^T(\tilde{\mathbf{z}}_j^* - \mathbf{A}\mathbf{c}_j) + n\mathbf{c}_j^T\mathbf{D}_{2,j}^{(m)}\mathbf{c}_j \Big\} \\
&= (\mathbf{I} + 8n\mathbf{D}_{2,j}^{(m)})^{-1}\mathbf{A}^T\tilde{\mathbf{z}}_j^*
\end{aligned} \tag{3.17}$$

and component-wise:

$$\hat{c}_{jl} = \frac{|c_{jl}^{(m)}|}{4nP'_\gamma(|c_{jl}^{(m)}|) + |c_{jl}^{(m)}|} h_{jl}, \quad j = 1, \cdots, d_2, \quad l = 1, \cdots, k, \tag{3.18}$$

where h_{jl} is the jlth entry of the matrix $\mathbf{H} = \mathbf{Z}^{*T}\mathbf{A}$. Alternating between (3.11), (3.12), (3.13), (3.14), (3.16) and (3.18) until convergence minimizes g to a local minimum. And the algorithmic details are summarized and presented in Algorithm 1. In that algorithm, we need to pre-specify the dimensionality k, which is the number of columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. In other words, the scores and loadings produced by Algorithm 1 depend on the value of k. Noticeably, our method is not a sequential method as the regular PCA is, instead, all the loadings and scores are obtained simultaneously. However, we do provide a way shortly to order the importance of the columns of the score and loading matrices so a layer-by-layer structure decreasing in importance can be obtained, giving rise to appealing interpretations.

After the convergence of Algorithm 1, each column of the estimated loading matrix $\mathbf{B}$ and $\mathbf{C}$ is made to have length unity and the score matrix $\mathbf{A}$ is re-estimated using Algorithm 3 with all the other parameters kept constant. Re-scaling the loading matrices bestows the interpretation of unit basis vectors on the columns of $\mathbf{B}$ and $\mathbf{C}$ as explained in Section 2.4.

Once the score matrix $\mathbf{A}$ is re-estimated with loadings $\mathbf{B}$ and $\mathbf{C}$ re-scaled, we consider ordering the columns of $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{A}$ consistently by some criterion to be discussed, such that we claim the first column of $\mathbf{B}$ and $\mathbf{C}$, after ordering, representing the set of variables in $\mathbf{X}$ with specific assigned weights and the set of variables in $\mathbf{Y}$ with specific assigned weights, are together of the most importance. Similarly, the second important pair, comprised of the second column of $\mathbf{B}$ and $\mathbf{C}$, is claimed, and so forth. Specifically, we propose to use the modified variance explained, developed by Shen and Huang (2008), as the criterion to come up with an order of importance. Suppose the first p most important columns of $\mathbf{B}$ or $\mathbf{C}$ are determined and their indices are $i_1, i_2, \cdots, i_p$, then the $(p+1)$th most important column of $\mathbf{B}$ or $\mathbf{C}$ indexed by i_{p+1} is decided as follows: We define $\mathbf{X}' = \mathbf{A}\mathbf{B}^T$ and $\mathbf{Y}' = \mathbf{A}\mathbf{C}^T$, and in turn, define $\mathbf{X}_{p+1} = \mathbf{X}'\mathbf{U}_{p+1}(\mathbf{U}_{p+1}^T\mathbf{U}_{p+1})^{-1}\mathbf{U}_{p+1}^T$ and $\mathbf{Y}_{p+1} = \mathbf{Y}'\mathbf{V}_{p+1}(\mathbf{V}_{p+1}^T\mathbf{V}_{p+1})^{-1}\mathbf{V}_{p+1}^T$, where $\mathbf{U}_{p+1}$ consists of the columns of $\mathbf{B}$ with indices $i_1, i_2, \cdots, i_p$ and i which is a running index rather than $i_1, i_2, \cdots, i_p$, indicating the $(p+1)$th column to be determined, by similarity, $\mathbf{V}_{p+1}$ is formed by stacking the columns of $\mathbf{C}$ with indices $i_1, i_2, \cdots, i_p$ and the same i. The index i_{p+1}, representing the $(p+1)$th most important column of $\mathbf{B}$ or $\mathbf{C}$ is decided by solving $i_{p+1} = \underset{i}{argmax}\ \left\{tr(\mathbf{X}_{p+1}^T\mathbf{X}_{p+1}) + tr(\mathbf{Y}_{p+1}^T\mathbf{Y}_{p+1})\right\}$. Then the vector $(i_1, i_2, \cdots, i_p, i_{p+1})$, containing the indices of the first $p+1$ most important columns of $\mathbf{B}$ or $\mathbf{C}$, constructs the first $p+1$ columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. In this way, a layer-by-layer structure as that of PCA or CCA is established, where an extra layer is the one, which is the projection directions, for both the continuous and the binary

data sets, defined in a particular column of **B** and **C**, bringing the most variability explained. Alternatively, as a great simplification, we can calculate the variances of each column of **A** and put them in a decreasing order, such that the column with the largest variance represents the first column of the ordered **A**, and the column with the second largest variance represents the second column, and so on. After the columns of **A** are ordered, the loading matrices **B** and **C** are ordered according to the column ordering of **A**. By this simplified way, we can plot the column variability (variance) of each column of **A** against its column index in a fashion of a scree plot, and, based on that, we can even decide the number of dominant layers to use when it comes to the interpretation of data analysis results. Again, the importance here is measured by the amount of information or variability shared in both **X** and **Y** that a particular pair of projection directions for each of the data sets could account for. Our empirical studies show the modified variance explained approach and the simplified approach usually agree with each other, especially on the order of the first few dominant layers, and hence we use the simplified one hereafter in the data analysis parts.

3.2 Estimation of α

This section discloses the technical details of the principle explained in Section 2.2. In order to extract the main signals contained in **X** alone by compressing it under penalty, the criterion function defined in (2.8) has to be minimized with optimally chosen λ, where a single λ for all the columns of $\mathbf{B}_X$ is used for simplicity. Employing the same techniques as used for minimizing (2.6) and dropping the subscript X for

notational elegance, we have, under L_1 penalty:

$$\begin{aligned} P_\lambda(\mathbf{B}) &= \lambda \sum_{j=1}^{d_1} \sum_{l=1}^{k} |b_{jl}| \\ &\leq \lambda \sum_{j=1}^{d_1} \sum_{l=1}^{k} \frac{b_{jl}^2}{2|b_{jl}^{(m)}|} + Cons. \\ &= \lambda \sum_{j=1}^{d_1} \mathbf{b}_j^T \mathbf{D}_{X,j}^{(m)} \mathbf{b}_j + Cons. \end{aligned} \tag{3.19}$$

and $\mathbf{D}_{X,j}^{(m)} = diag\Big(\frac{1}{2|b_{j1}^{(m)}|}, \cdots, \frac{1}{2|b_{jk}^{(m)}|}\Big)$. In turn, $S_X(\mu_X, \mathbf{A}_X, \mathbf{B}_X, \sigma_X^2)$ in (2.8) is bounded as:

$$\begin{aligned} S(\mu, \mathbf{A}, \mathbf{B}, \sigma^2) \leq \frac{nd_1}{2} log\sigma^2 + \frac{1}{2\sigma^2} \sum_{j=1}^{d_1} \sum_{i=1}^{n} \left(x_{ij} - (\mu_j + \mathbf{a}_i^T \mathbf{b}_j)\right)^2 \\ + \lambda n \sum_{j=1}^{d_1} \mathbf{b}_j^T \mathbf{D}_{X,j}^{(m)} \mathbf{b}_j + Cons. \end{aligned} \tag{3.20}$$

Minimizing the majorizing function on the right hand side of the above inequality can be solved iteratively by alternating the minimization with respect to μ, σ^2, $\mathbf{A}$ and $\mathbf{B}$ as how the function g is minimized in the previous section. We skip the derivation and present the algorithmic procedures in Algorithm 4. The search for the best tuning parameter λ utilizes the BIC defined as:

$$BIC(\lambda) = -2l(\mu, \mathbf{A}, \mathbf{B}, \sigma^2) + log\, n \times m_X(\lambda) \tag{3.21}$$

and $m_X(\lambda)$, a measure of the degrees of freedom for the log-likelihood of $\mathbf{X}$ alone, is defined as $m_X(\lambda) = d_1 + nk + |\mathfrak{B}(\lambda)|$, where d_1 is the length of vector μ, nk is the total number of elements in A, and $|\mathfrak{B}(\lambda)|$ is the cardinality of the index set $\mathfrak{B}(\lambda)$ of the nonzero entries in $\mathbf{B}$ when the penalty parameter is λ. On the other

hand, minimization of (2.11) needs to be solved in order to extract the main signals contained in $\mathbf{Y}$ with optimally chosen tuning parameter γ which is used for all the columns of $\mathbf{C}_Y$. The subscript Y is dropped whenever there's no ambiguity in the rest of this section. Under L_1 penalty:

$$\begin{aligned} P_\gamma(\mathbf{C}) &= \gamma \sum_{j=1}^{d_2} \sum_{l=1}^{k} |c_{jl}| \\ &\leq \gamma \sum_{j=1}^{d_2} \mathbf{c}_j^T \mathbf{D}_{Y,j}^{(m)} \mathbf{c}_j + Cons. \end{aligned} \tag{3.22}$$

and $\mathbf{D}_{Y,j}^{(m)} = diag\Big(\frac{1}{2|c_{j1}^{(m)}|}, \cdots, \frac{1}{2|c_{jk}^{(m)}|}\Big)$. Moreover, the negative log-likelihood of $\mathbf{Y}$, $-l_Y(\nu_Y, \mathbf{A}_Y, \mathbf{C}_Y)$, defined in (2.10) is bounded by:

$$-l(\nu, \mathbf{A}, \mathbf{C}) \leq \sum_{j=1}^{d_2} \sum_{i=1}^{n} \frac{1}{8} \big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T \mathbf{c}_j)\big)^2 + Cons., \tag{3.23}$$

where $z_{ij}^{(m)} = \theta_{ij}^{(m)} + 4q_{ij}\big(1 - \pi(q_{ij}\theta_{ij}^{(m)})\big)$ and $\theta_{ij}^{(m)}$ is defined the same as it is in the main algorithm. Therefore, $S_Y(\nu_Y, \mathbf{A}_Y, \mathbf{C}_Y)$ in (2.11) is majorized as:

$$S(\nu, \mathbf{A}, \mathbf{C}) \leq \sum_{j=1}^{d_2} \sum_{i=1}^{n} \frac{1}{8} \big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T \mathbf{c}_j)\big)^2 + \gamma n \sum_{j=1}^{d_2} \mathbf{c}_j^T \mathbf{D}_{Y,j}^{(m)} \mathbf{c}_j + Cons. \tag{3.24}$$

Iteratively alternating minimizing the right hand side of the above inequality with respect to ν, $\mathbf{A}$ and $\mathbf{C}$ produces a solution to the minimization problem in (2.11) and the detailed procedures are shown in Algorithm 5. The optimal tuning parameter γ can be determined again by BIC:

$$BIC(\gamma) = -2l(\nu, \mathbf{A}, \mathbf{C}) + log\ n \times m_Y(\gamma) \tag{3.25}$$

and $m_Y(\gamma)$ is a measure of the degrees of freedom for the log-likelihood of $\mathbf{Y}$ alone, defined as $m_Y(\gamma) = d_2 + nk + |\mathfrak{C}(\gamma)|$, where d_2 is the length of vector ν, nk is the total number of elements in A, and $|\mathfrak{C}(\gamma)|$ is the cardinality of the index set $\mathfrak{C}(\gamma)$ of the nonzero entries in $\mathbf{C}$ when the penalty parameter is γ. After the minimization problem in (2.8) and (2.11) are solved with optimal tuning parameters respectively, the balancing parameter α can be calculated straightforwardly as described in Section 2.2.

3.3 Initialization to the Main Algorithm

In general, an MM algorithm can only guarantee a convergence to a local minimum as characterized in many other nonlinear optimization algorithms. In search of a global minimum, one generally accepted practice is to kick off the algorithm many times randomly at different starting points and find the best one according to some criterion. However, random initializations could be instable, let alone consuming more time for convergence, and multiple trials could be inefficient. As a remedy, we propose a deterministic way, borrowing the ideas of the singular value decomposition (SVD), to initialize our main algorithm (Algorithm 1). The procedures are listed below:

1. Extract μ_X, $\mathbf{A}_X$ and $\mathbf{B}_X$ by applying Algorithm 4 to data set $\mathbf{X}$ with optimally chosen λ as described in Section 3.2.

2. Extract ν_Y, $\mathbf{A}_Y$ and $\mathbf{C}_Y$ by applying Algorithm 5 to data set $\mathbf{Y}$ with optimally chosen γ as described in Section 3.2.

3. Calculate the demeaned features of $\mathbf{X}$ defined as $\mathbf{X}_d = \mathbf{A}_X\mathbf{B}_X^T$.

4. Calculate the demeaned features of $\mathbf{Y}$ defined as $\mathbf{Y}_d = \mathbf{A}_Y\mathbf{C}_Y^T$.

5. Form the augmented matrix by stacking up the columns of $\mathbf{X}_d$ and $\mathbf{Y}_d$, as $\mathbf{AUG} = \big[\mathbf{X}_d|\mathbf{Y}_d\big]$.

6. Center and standardize the augmented matrix column-wise. If missing values occur, replace the missing values by the corresponding entries of the augmented matrix before centering and standardization.

7. Decompose the centered and standardized augmented matrix by SVD as $\mathbf{UDV}^T$.

8. For the initialization to the main algorithm, μ is initialized by μ_X; ν is initialized by ν_Y; $\mathbf{A}$ is initialized by $\mathbf{UD}$; $\mathbf{B}$ is initialized by the first d_1 rows of $\mathbf{V}$ and $\mathbf{C}$ is initialized by the last d_2 rows of $\mathbf{V}$.

The idea of the above procedures is to combine the optimally estimated main signals in $\mathbf{X}$ and $\mathbf{Y}$, so the initial value of $\mathbf{A}$ to the main algorithm can benefit from information sharing between both data sets given the interpretation of SVD. Our simulation studies show this initializing approach works very well in terms of boosting up the estimation quality of the main algorithm and reducing the number of loops required before convergence.

Algorithm 1: Parameter Estimation with SCAD Penalty on Loadings

1. **Initialization**
   Initialize $\mu^{(0)}, \nu^{(0)}, \mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}$ and set $\sigma^{2(0)} = 1$, $\mathbf{M}^{(0)} = \mathbf{0}_{n \times d_1}$.
   Set $m = 0$.

2. **Update** μ
   Set $\mathbf{X}^{\dagger(m)} = \left(x_{ij}^{\dagger(m)}\right)$ with $x_{ij}^{\dagger(m)} = x_{ij} - \mathbf{a}_i^{(m)T}\mathbf{b}_j^{(m)}$. Update μ using $\mu^{(m+1)} = \frac{1}{n}\mathbf{X}^{\dagger(m)T}\mathbf{1}_n$.

3. Compute $z_{ij}^{(m)} = \theta_{ij}^{(m)} + 4q_{ij}\big(1 - \pi(q_{ij}\theta_{ij}^{(m)})\big)$, where $\mathbf{\Theta}^{(m)} = \mathbf{1_n} \otimes \nu^{(m)T} + \mathbf{A}^{(m)}\mathbf{C}^{(m)T}$, and set $\mathbf{Z}^{(m)} = \left(z_{ij}^{(m)}\right)$.

4. **Update** ν
   Set $\mathbf{Z}^{\dagger(m)} = \left(z_{ij}^{\dagger(m)}\right)$ with $z_{ij}^{\dagger(m)} = z_{ij}^{(m)} - \mathbf{a}_i^{(m)T}\mathbf{c}_j^{(m)}$. Update ν using $\nu^{(m+1)} = \frac{1}{n}\mathbf{Z}^{\dagger(m)T}\mathbf{1}_n$.

5. **Update** σ^2
   Update σ^2 using $(\sigma^2)^{(m+1)} = \frac{1}{nd_1} tr(\mathbf{M}^{(m+1)}\mathbf{M}^{(m+1)T})$,
   where $\mathbf{M}^{(m+1)} = \mathbf{X}^{\dagger(m)} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.

6. **Inner iterations** (See Algorithm 2)

   a. **Update A**

   b. **Update B**

   c. **Update C**

7. Repeat 2. - 6. with $m = m + 1$ until convergence.

Algorithm 2: Parameter Estimation with SCAD Penalty on Loadings (Inner Iterations)

a. **Update A**
Set $\mathbf{Z}^{*(m+1)} = \mathbf{Z}^{(m)} - \mathbf{1}_n \otimes \nu^{(m+1)T}$ and $\mathbf{X}^{*(m+1)} = \mathbf{X} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.
$\mathbf{A}^{(m+1)} = \left(\frac{1}{4}\mathbf{Z}^{*(m+1)}\mathbf{C}^{(m)} + \frac{\alpha}{(\sigma^2)^{(m+1)}}\mathbf{X}^{*(m+1)}\mathbf{B}^{(m)}\right)\left(\frac{1}{4}\mathbf{C}^{(m)T}\mathbf{C}^{(m)} + \frac{\alpha}{(\sigma^2)^{(m+1)}}\mathbf{B}^{(m)T}\mathbf{B}^{(m)}\right)^{-1}$.

Compute the QR decomposition $\mathbf{A}^{(m+1)} = \mathbf{QR}$,
and then replace $\mathbf{A}^{(m+1)}$ by $\mathbf{Q}$.

b. **Update B**
Set $\mathbf{G}^{(m+1)} = \left(g_{jl}^{(m+1)}\right) = (\mathbf{X}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
Update $\mathbf{B}$ by $\mathbf{B}^{(m+1)} = \left(b_{jl}^{(m+1)}\right)$,
where $b_{jl}^{(m+1)} = \dfrac{\alpha|b_{jl}^{(m)}|}{n(\sigma^2)^{(m+1)}P'_\lambda(|b_{jl}^{(m)}|) + \alpha|b_{jl}^{(m)}|}g_{jl}^{(m+1)}$,
$l = 1, \cdots, k$ and $j = 1, \cdots, d_1$.

c. **Update C**
Set $\mathbf{H}^{(m+1)} = \left(h_{jl}^{(m+1)}\right) = (\mathbf{Z}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
Update $\mathbf{C}$ by $\mathbf{C}^{(m+1)} = \left(c_{jl}^{(m+1)}\right)$,
where $c_{jl}^{(m+1)} = \dfrac{|c_{jl}^{(m)}|}{4nP'_\gamma(|c_{jl}^{(m)}|) + |c_{jl}^{(m)}|}h_{jl}^{(m+1)}$,
$l = 1, \cdots, k$ and $j = 1, \cdots, d_2$.

Algorithm 3: Post-convergence Procedures

Input: μ, ν, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ estimated from Algorithm 1.

1. Make each column of $\mathbf{B}$ and $\mathbf{C}$ have length unity.
2. Calculate $\mathbf{\Theta} = \mathbf{1}_n \otimes \nu^T + \mathbf{A}\mathbf{C}^T$.
3. Compute $\mathbf{Z}$, where $z_{ij} = \theta_{ij} + 4q_{ij}\big(1 - \pi(q_{ij}\theta_{ij})\big)$.
4. Compute $\mathbf{X}^\dagger = \mathbf{X} - \mathbf{A}\mathbf{B}^T$.
5. Calculate $\mathbf{M} = \mathbf{X}^\dagger - \mathbf{1}_n \otimes \mu^T$.
6. Calculate $\sigma^2 = \dfrac{1}{nd_1} tr(\mathbf{M}\mathbf{M}^T)$.
7. Compute $\mathbf{X}^* = \mathbf{X} - \mathbf{1}_n \otimes \mu^T$ and $\mathbf{Z}^* = \mathbf{Z} - \mathbf{1}_n \otimes \nu^T$.
8. Calculate $\mathbf{A} = \left(\dfrac{1}{4}\mathbf{Z}^*\mathbf{C} + \dfrac{\alpha}{\sigma^2}\mathbf{X}^*\mathbf{B}\right)\left(\dfrac{1}{4}\mathbf{C}^T\mathbf{C} + \dfrac{\alpha}{\sigma^2}\mathbf{B}^T\mathbf{B}\right)^{-1}$.
9. Repeat step 2. to 8. until convergence.

Algorithm 4: Sparse Probabilistic PCA with L_1 Penalty

Input: $\mathbf{X}_{n\times d_1}$ and k: # of loadings

1. **Initialization**
   Initialize $\mu^{(0)}, \mathbf{A}^{(0)}, \mathbf{B}^{(0)}$ and set $\sigma^{2(0)} = 1$, $\mathbf{M}^{(0)} = \mathbf{0}_{n\times d_1}$. Set $m = 0$.

2. **Update** μ
   Set $\mathbf{X}^{\dagger(m)} = \left(x_{ij}^{\dagger(m)}\right)$ with $x_{ij}^{\dagger(m)} = x_{ij} - \mathbf{a}_i^{(m)T}\mathbf{b}_j^{(m)}$. Update μ using $\mu^{(m+1)} = \dfrac{1}{n}\mathbf{X}^{\dagger(m)T}\mathbf{1}_n$.

3. **Update** σ^2
   Update σ^2 using $(\sigma^2)^{(m+1)} = \dfrac{1}{nd_1}tr(\mathbf{M}^{(m+1)}\mathbf{M}^{(m+1)T})$,
   where $\mathbf{M}^{(m+1)} = \mathbf{X}^{\dagger(m)} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.

4. **Update A**
   Set $\mathbf{X}^{*(m+1)} = \mathbf{X} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.
   Then $\mathbf{A}^{(m+1)} = \mathbf{X}^{*(m+1)}\mathbf{B}^{(m)}\left(\mathbf{B}^{(m)T}\mathbf{B}^{(m)}\right)^{-1}$.

   Compute the QR decomposition $\mathbf{A}^{(m+1)} = \mathbf{QR}$,
   and then replace $\mathbf{A}^{(m+1)}$ by $\mathbf{Q}$.

5. **Update B**
   Set $\mathbf{G}^{(m+1)} = \left(g_{jl}^{(m+1)}\right) = (\mathbf{X}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
   Update $\mathbf{B}$ by $\mathbf{B}^{(m+1)} = \left(b_{jl}^{(m+1)}\right)$,
   where $b_{jl}^{(m+1)} = \dfrac{|b_{jl}^{(m)}|}{\lambda n(\sigma^2)^{(m+1)} + |b_{jl}^{(m)}|}g_{jl}^{(m+1)}$,
   $l = 1, \cdots, k$ and $j = 1, \cdots, d_1$.

6. Repeat 2. - 5. with $m = m + 1$ until convergence.

Algorithm 5: Sparse Logistic PCA with L_1 Penalty

Input: $\mathbf{Y}_{n\times d_2}$ and k: # of loadings

1. **Initialization**
   Initialize $\nu^{(0)}, \mathbf{A}^{(0)}, \mathbf{C}^{(0)}$ and set $m = 0$.

2. Compute $z_{ij}^{(m)} = \theta_{ij}^{(m)} + 4q_{ij}\big(1 - \pi(q_{ij}\theta_{ij}^{(m)})\big)$, where
   $\mathbf{\Theta}^{(m)} = \mathbf{1_n} \otimes \nu^{(m)T} + \mathbf{A}^{(m)}\mathbf{C}^{(m)T}$, and set $\mathbf{Z}^{(m)} = \big(z_{ij}^{(m)}\big)$.

3. **Update ν**
   Set $\mathbf{Z}^{\dagger(m)} = \big(z_{ij}^{\dagger(m)}\big)$ with $z_{ij}^{\dagger(m)} = z_{ij}^{(m)} - \mathbf{a}_i^{(m)T}\mathbf{c}_j^{(m)}$. Update ν using
   $\nu^{(m+1)} = \dfrac{1}{n}\mathbf{Z}^{\dagger(m)T}\mathbf{1}_n$.

4. **Update A**
   Set $\mathbf{Z}^{*(m+1)} = \mathbf{Z}^{(m)} - \mathbf{1}_n \otimes \nu^{(m+1)T}$. Then
   $\mathbf{A}^{(m+1)} = \mathbf{Z}^{*(m+1)}\mathbf{C}^{(m)}\big(\mathbf{C}^{(m)T}\mathbf{C}^{(m)}\big)^{-1}$.

   Compute the QR decomposition $\mathbf{A}^{(m+1)} = \mathbf{QR}$,
   and then replace $\mathbf{A}^{(m+1)}$ by $\mathbf{Q}$.

5. **Update C**
   Set $\mathbf{H}^{(m+1)} = \big(h_{jl}^{(m+1)}\big) = (\mathbf{Z}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
   Update $\mathbf{C}$ by $\mathbf{C}^{(m+1)} = \big(c_{jl}^{(m+1)}\big)$,
   where $c_{jl}^{(m+1)} = \dfrac{|c_{jl}^{(m)}|}{4\gamma n + |c_{jl}^{(m)}|}h_{jl}^{(m+1)}$,
   $l = 1, \cdots, k$ and $j = 1, \cdots, d_2$.

6. Repeat 2. - 5. with $m = m + 1$ until convergence.

4. SIMULATION STUDIES

In the simulation section, we demonstrate the effectiveness and advantages of our proposed method using five sets of simulation studies presented in two subsections. In 4.1, we generate four sets of data sets with different combinations of dimensionality and number of subjects, from the model as described in Section 2 to test the effectiveness and stability of the proposed algorithms. In the meantime, we compare the results of our method with those produced by some off-the-shelf sparse canonical correlation analysis (SCCA) methods as some authors proposed variants of SCCA to handle eQTL mapping tasks, although CCA originally was not designed specifically for that purpose. However, we need to point out that CCA is more suitable to be carried out on continuous variables for its natural interpretation. In section 4.2, our method is applied to two other data sets with intrinsic correlation relationship not generated directly from the model. The purpose of that is to see if the method proposed could capture those internal relationships and illustrate its interpretability at the same time.

4.1 Simulation with Data Generated from the Model

In this subsection, we have four simulation studies with two dimensionality setups and two sample size setups, making totally four possible combinations. For each one of the studies, the continuous data set and the binary data set are generated directly from the model in Section 2 according to equation (2.1) and (2.3). As mentioned, the purpose is to see if the proposed method is able to reflect the true structure embedded in the data sets reasonably well. Based on our empirical studies, our method works very well for data sets by various settings where the number of observations is greater than the number of variables, that is, $n > d_1$ and $n > d_2$. But here we present a

challenging case in which $d_1 > n$ and $d_2 > n$, and particularly, we make d_1 and d_2, the dimension of the continuous and the binary data set respectively, much larger than n, the number of observations, mimicking data obtained from microarray analysis and data sets with genetic markers. In the mean time, we compare with the performances of two off-the-shelf SCCA methods, the regularized CCA (rCCA) by González et al. (2008) and a penalized matrix decomposition (PMD) by Witten et al. (2009), as they are recently developed and effective methods with software implementation in R package 'mixOmics' and 'PMA' respectively. As mentioned above, our method is designed with a different purpose than sparse CCA and hence bears different interpretations. However, it is still of interest to see, to what extent, those two sparse CCA methods could uncover the underlying signals since sparse CCA methods are indeed actually used for eQTL mapping purposes.

4.1.1 Study with Moderate Dimensionality and Small Sample Size (Simulation 1).

The continuous data set $\mathbf{X}$ and the binary data set $\mathbf{Y}$ are generated from the model as described by equation (2.1) and (2.3). Recall that matrix $\mathbf{A}$ is of dimension $n \times k$, matrix $\mathbf{B}$ of dimension $d_1 \times k$, and $\mathbf{C}$ of dimension $d_2 \times k$. Set $n = 30$ (small sample size), the true intrinsic dimension of both data sets $k = 10$, $d_1 = 2000$ and $d_2 = 200$. Matrix $\mathbf{A}$ is generated column-wise. For the first column of $\mathbf{A}$, the first 15 entries follow a normal distribution with mean 20 and variance 1 and the rest 15 entries are drawn from a normal with mean -20 and variance 1. From the second column of $\mathbf{A}$ to the tenth, each column individually follows a normal distribution with mean zero and a particular variance. And the variances of the columns (2 to 10) of $\mathbf{A}$ are determined as the following: $diag(Var(\mathbf{A}_{sub})) = ratio \times base$, where $\mathbf{A}_{sub}$ is the submatrix comprised of all the columns of $\mathbf{A}$ except for the first, *base* is set to be 50 and *ratio* is a vector of length 9 and in this simulation

$ratio = (6, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01)$. For example, the variance of the second column of $\mathbf{A}$ is $50 \times 6 = 300$, and the variance of the third column of $\mathbf{A}$ is $50 \times 0.01 = 0.5$, and so forth. With this setup of $\mathbf{A}$, we naturally embed grouping information, that is, observations evenly assigned to two groups, in the first column, and we make the column variances decreasing with the first two columns dominant. The variances of the columns of the true $\mathbf{A}$ generated are plotted in the left panel of Figure 4.1. As seen from the figure, we should expect the first two layers to capture the main signals. The grouping information is depicted in the right panel of Figure 4.1, where the second column of true $\mathbf{A}$ is plotted against the first. For the generation of true $\mathbf{B}_{2000\times10}$, the first column sequentially has 1940 0's, 20 -1's, 20 0's and 20 -1's; the second column has 1920 0's, 20 1's, 20 0's, 20 1's and 20 0's; for column 3 to column 10, 10% of the entries are set at random to 1 and the rest are set to 0. And then each column of the generated true $\mathbf{B}$ is re-scaled to have length unity. Figure 4.2 depicts the first two columns of the true $\mathbf{B}$ generated. The true $\mathbf{C}_{200\times10}$ is determined as the following: The first column is sequentially comprised of 20 1's, 20 0's, 20 1's and 140 0's; the second column has 20 0's, 20 -1's, 20 0's, 20 -1's and 120 0's; for column 3 to column 10, 10% of the entries are randomly set to 1 and the rest are set to 0. And then each column of the generated true $\mathbf{C}$ is re-scaled to have length unity. The plots of the first two columns of the true $\mathbf{C}$ can be found in Figure 4.3. Besides, μ and ν are set to be zero vectors, and σ^2, the variance of the errors, is set to equal 1. Then, matrix $\mathbf{X}$ can be formed by equation (2.1) with μ, $\mathbf{A}$, $\mathbf{B}$ and errors generated. On the other hand, matrix $\mathbf{Y}$ is generated as follows: Form matrix $\mathbf{\Theta}$ by equation (2.3) with ν, $\mathbf{A}$ and $\mathbf{C}$ given and y_{ij} is generated from a Bernoulli with success probability $\pi_{ij} = \pi(\theta_{ij})$ independently.

We note that we set $k = 10$ in the algorithms during the estimation process to make the results more comparable. From the left panel of Figure 4.4, we can see

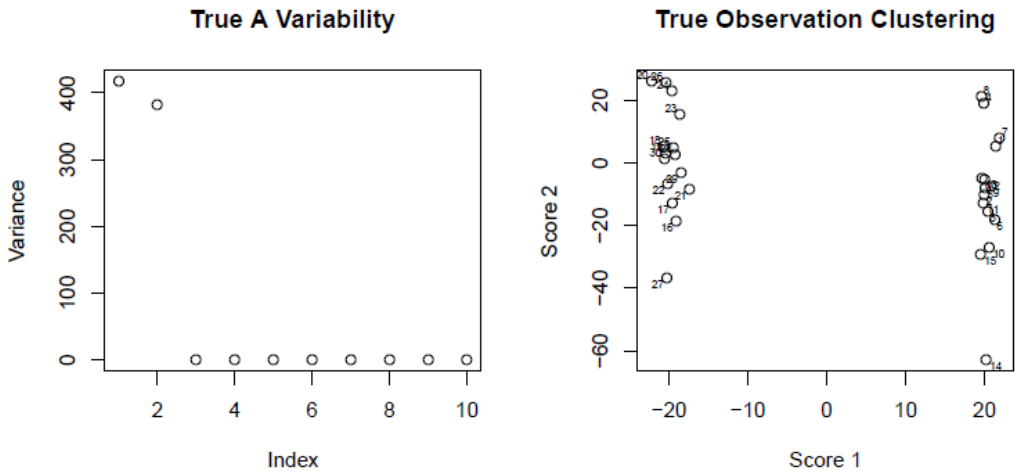

Figure 4.1: Variability of true A and true observational clustering (Simulation 1)

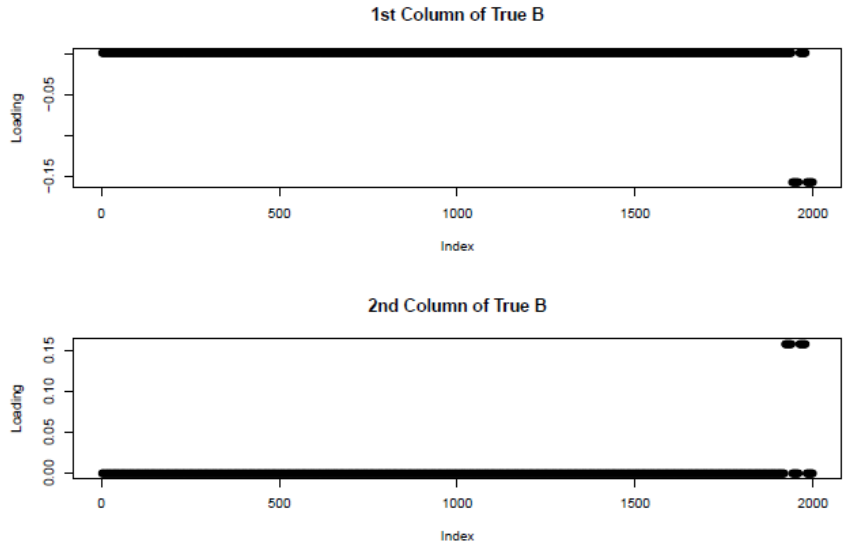

Figure 4.2: The first two columns of true B (Simulation 1)

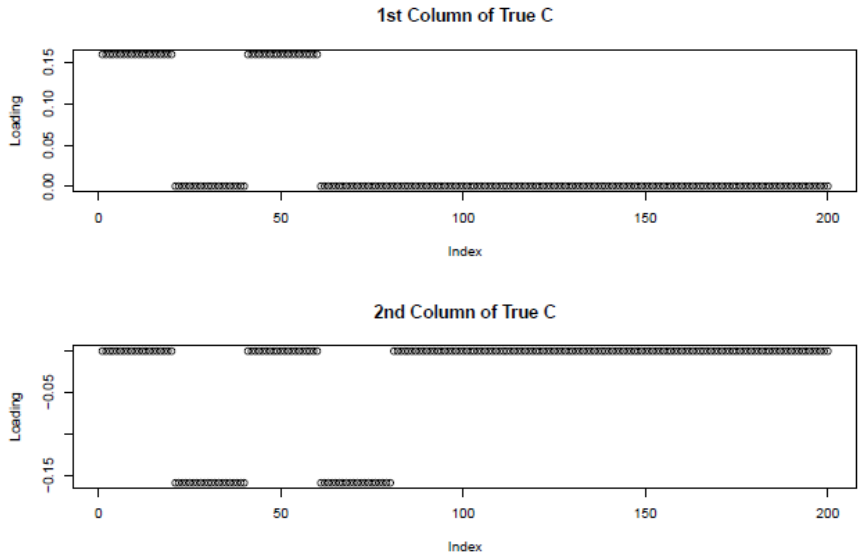

Figure 4.3: The first two columns of true C (Simulation 1)

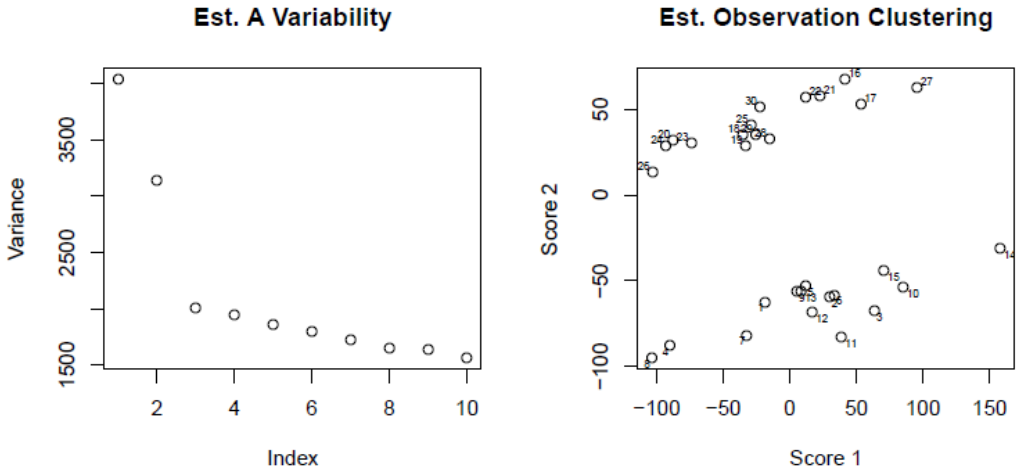

Figure 4.4: Variability of estimated A and estimated observational clustering (Simulation 1)

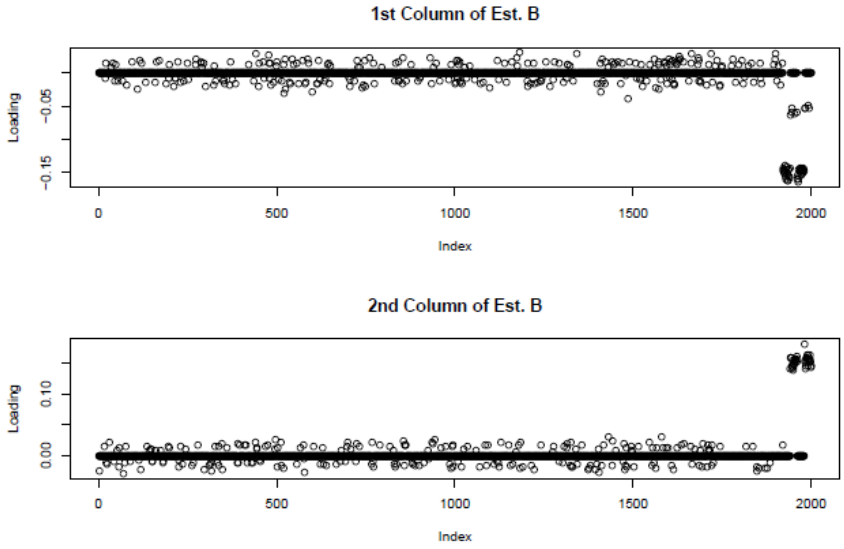

Figure 4.5: The first two columns of estimated B (Simulation 1)

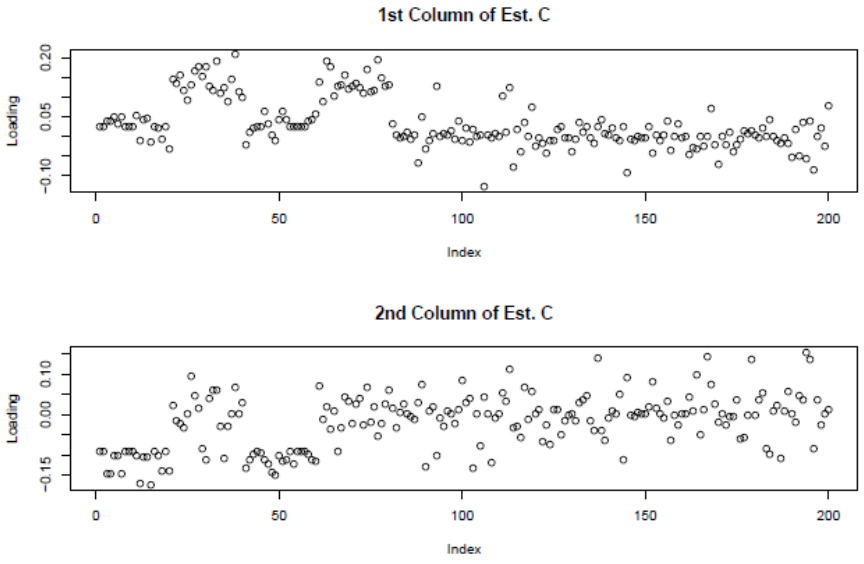

Figure 4.6: The first two columns of estimated C (Simulation 1)

the two most dominant layers are captured by our algorithms and the embedded grouping information captured by the second layer of the estimated $\mathbf{A}$ is shown on the right. The corresponding columns of the estimated loading matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ are displayed in Figure 4.5 and 4.6, respectively. If we compare the first two columns of $\hat{\mathbf{B}}$ to the first two columns of true $\mathbf{B}$ and the first two columns of $\hat{\mathbf{C}}$ to those of true $\mathbf{C}$, we observe the estimated version has the first and the second layers switched possibly due to a rotation but the majority of true signals are able to stand out with accurate magnitude, except for differing from the truth only by signs and with some interfering noises. The variability in the estimated plots is expected as data generation using Bernoulli's introduces extra noises and uncertainty. If we focus on the big picture, we can safely conclude our algorithms honestly capture the signals, in terms of both locations and magnitude. As comparisons, we run the regularized CCA (rCCA) method and a penalized matrix decomposition (PMD) algorithm on the generated data sets with optimally chosen tuning parameters and we treat the binary data set $\mathbf{Y}$ as if it were continuous for the implementation of those two methods. We present the estimated leading loadings by the regularized CCA (rCCA) in Figure 4.7 and 4.8. We note here the interpretation of the loadings of $\mathbf{X}$ resembles that of $\mathbf{B}$ in our model and the meaning of the loadings of $\mathbf{Y}$ is similar to that of $\mathbf{C}$. From the panels of Figure 4.7, the locations of the true signals in true $\mathbf{B}$ are captured but the magnitude is shrunk by around 95%, let alone the non-negligible noises. It is also noticeable that the loadings suffer from under-regularization, especially seen from the loadings of $\mathbf{X}$. The locations of the true signals in the first two layers of $\mathbf{C}$ are reflected well, although the magnitude of the signals is not accurately captured by rCCA as shown in Figure 4.8. On the other hand, the clustering information can be obtained by studying the scores of $\mathbf{X}$ and $\mathbf{Y}$ separately because there is no unified definition of scores as the role $\mathbf{A}$ does in our model setting, and the plots are

displayed in Figure 4.9, indicating successful identification of clustering information. Applying PMD to the data sets, Figure 4.10 and 4.11 picture the first two estimated loadings of $\mathbf{X}$ and $\mathbf{Y}$ obtained with optimally chosen tuning parameters. As seen from the plots, the loadings of both $\mathbf{X}$ and $\mathbf{Y}$ are very heavily over-penalized resulting in significant amount of signal loss. And the observation clustering is depicted in Figure 4.12 by scores of $\mathbf{X}$ and $\mathbf{Y}$, separately. As a complement, the accuracy of the estimated loading vectors can also be measured quantitatively by principal angles (Miao and Ben-Israel, 1992). For instance, in order to measure the closeness of the first two columns of the estimated loading matrix $\hat{\mathbf{B}}$ and the corresponding columns of the true loading matrix $\mathbf{B}$, we use the principal angle between spaces spanned by the first two columns of $\hat{\mathbf{B}}$ and those of $\mathbf{B}$. As a result, by averaging on 50 replications, the principal angle between the space spanned by the first two columns of $\hat{\mathbf{B}}$ and the space spanned by the corresponding columns of true $\mathbf{B}$ is 18.75° (2.78°, standard deviation); the principal angle between the space spanned by the first two columns of $\hat{\mathbf{C}}$ and the column space of the first two columns of $\mathbf{C}$ is 40.53° (2.95°) by our approach. For rCCA, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is 43.89° (3.27°); the principal angle between the space spanned by the first two loading vectors of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is 27.48° (2.75°). Similarly, by PMD, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is 80.04° (6.12°); the principal angle between the space spanned by the first two loadings of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is 26.06° (16.13°). Here we need to put emphasis on a very important distinction of PMD from our approach and rCCA. In the process of tuning parameter selection of PMD, the 10-fold cross-validation procedure involves missing a non-overlapping one-tenth of

the elements of $\mathbf{X}^T\mathbf{Y}$, sampled at random from the rows and columns, for each fold. That means there is a lot of randomness in the course of tuning parameter selection, resulting, highly likely, in different pairs of tuning parameters selected. And the differences in the tuning parameters for each run might be quite noticeable due to different randomness patterns, which, in turn, will result in different estimation qualities. As contrast, our approach and rCCA method are deterministic as both of them involve no randomness at all during either the tuning process or the estimation process.

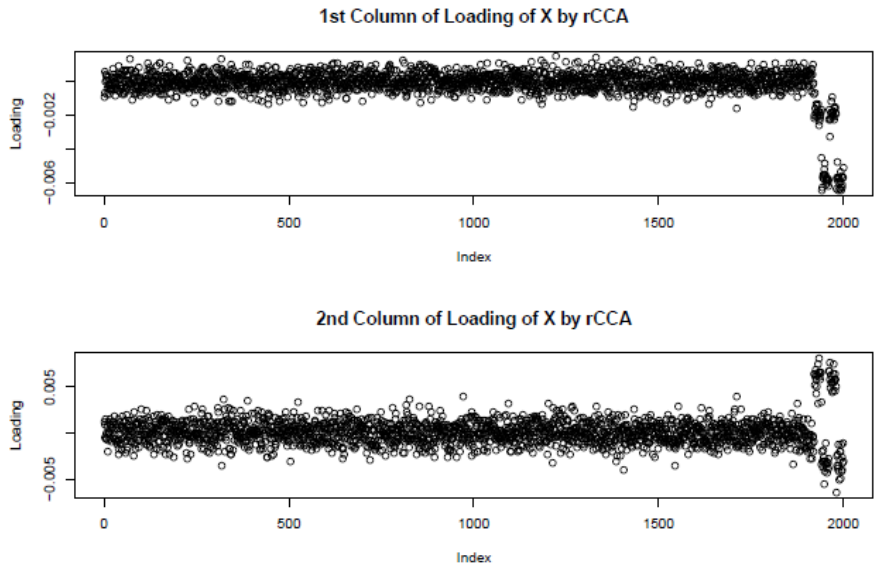

Figure 4.7: The first two columns of loadings of X by rCCA (Simulation 1)

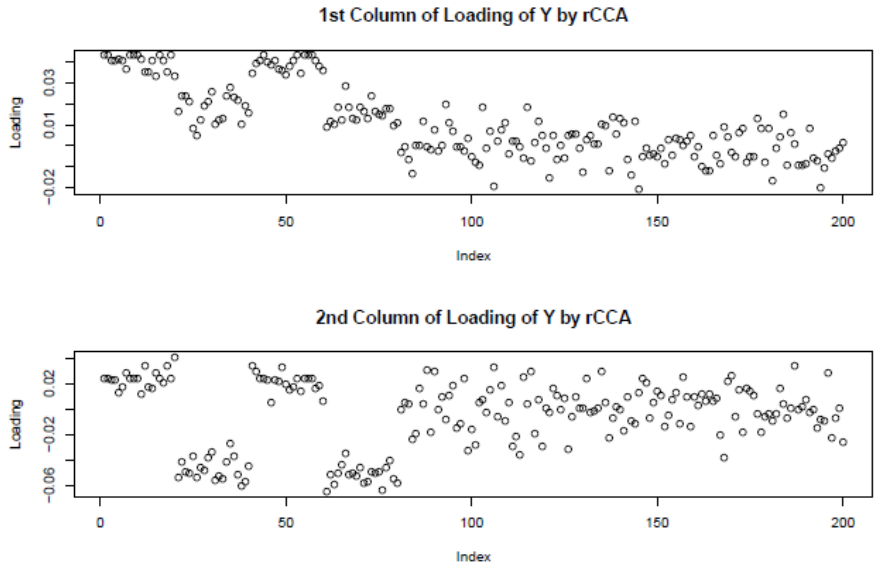

Figure 4.8: The first two columns of loadings of Y by rCCA (Simulation 1)

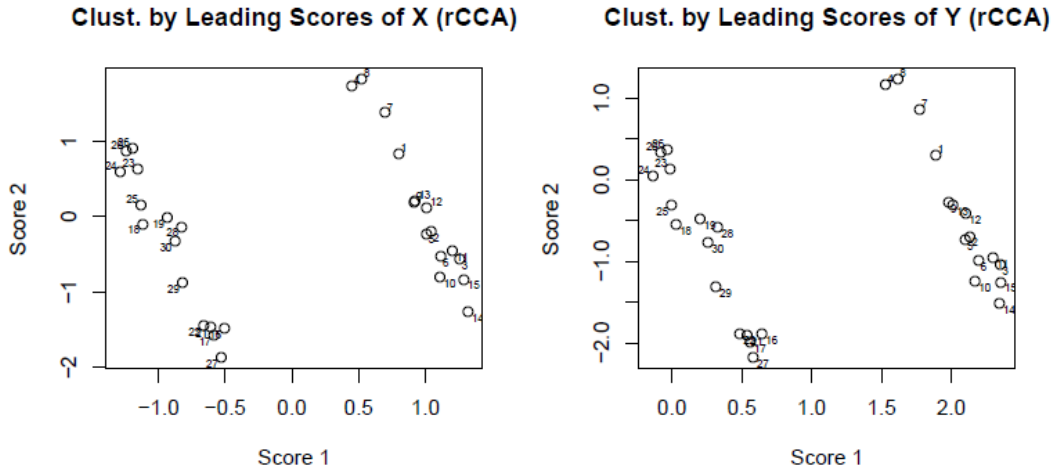

Figure 4.9: Clustering by leading scores of X and Y by rCCA (Simulation 1)

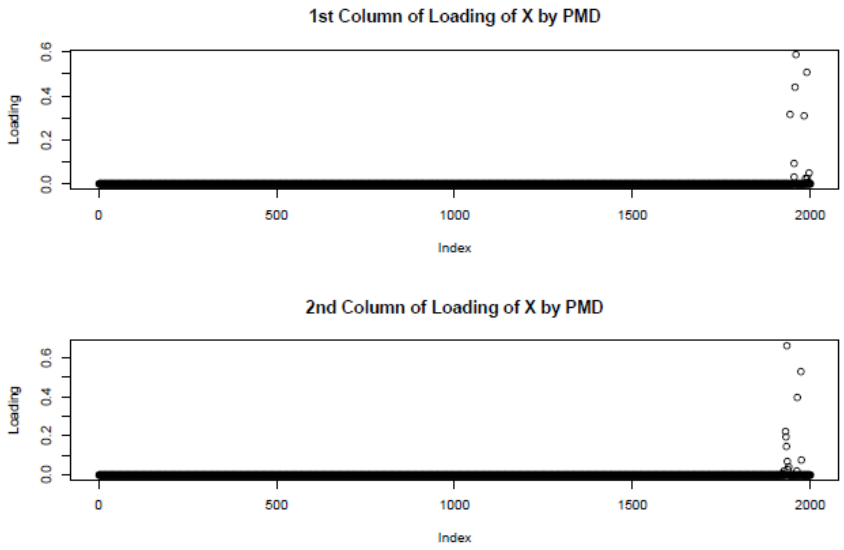

Figure 4.10: The first two columns of loadings of X by PMD (Simulation 1)

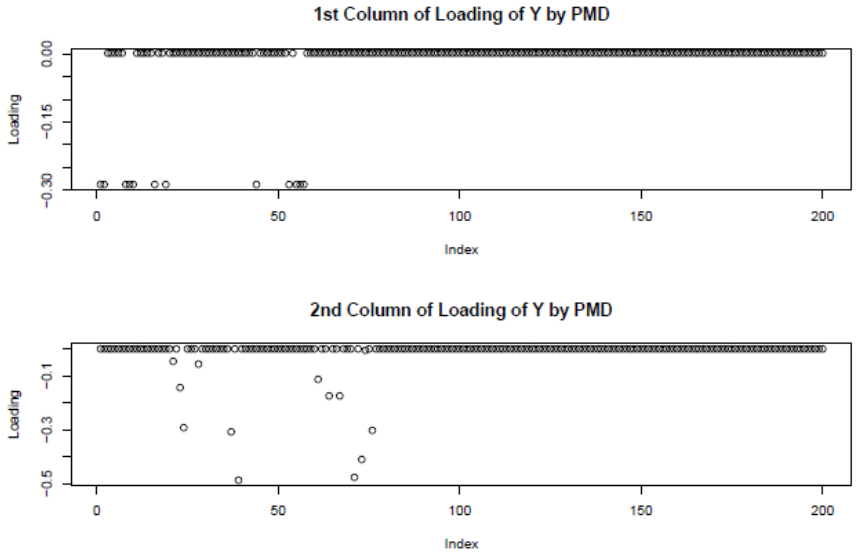

Figure 4.11: The first two columns of loadings of Y by PMD (Simulation 1)

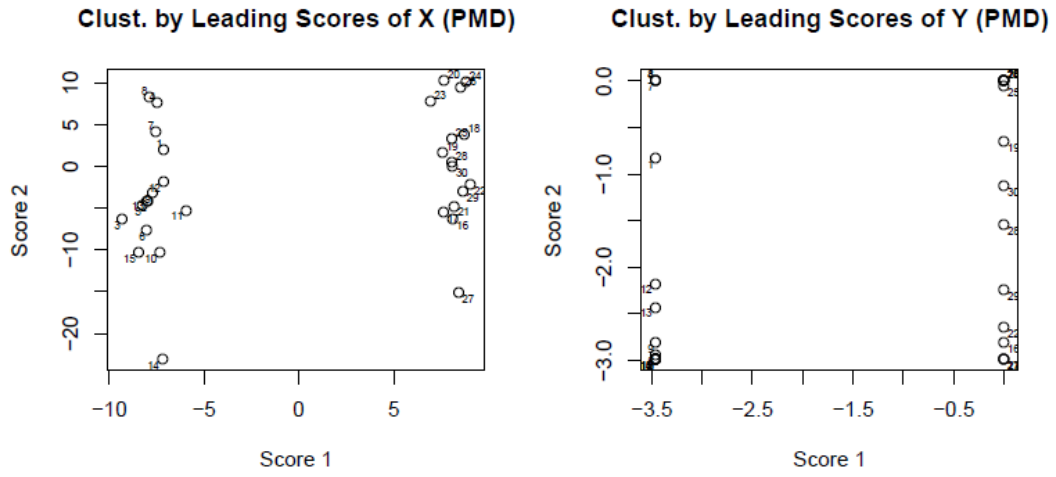

Figure 4.12: Clustering by leading scores of X and Y by PMD (Simulation 1)

4.1.2 Study with Moderate Dimensionality and Large Sample Size (Simulation 2).

In this simulation study, we compare the performances of rCCA and PMD with our method in a more favorable case where we increase the sample size to $n = 60$ and keep everything else the same. Particularly, we have $k = 10$, $d_1 = 2000$ and $d_2 = 200$. Matrix $\mathbf{A}$ is generated column-wise exactly as how it is done in the previous simulation except the first column assigns the first 30 subjects to a group and the rest 30 to the other. The variances of the columns of the true $\mathbf{A}$ generated are plotted in the left panel of Figure 4.13 and the clustering information is contained in the right panel. The loadings $\mathbf{B}$ and $\mathbf{C}$ are set exactly the same as they are in Simulation 1 and the plots of their leading columns can be found in Figure 4.2 and 4.3 respectively.

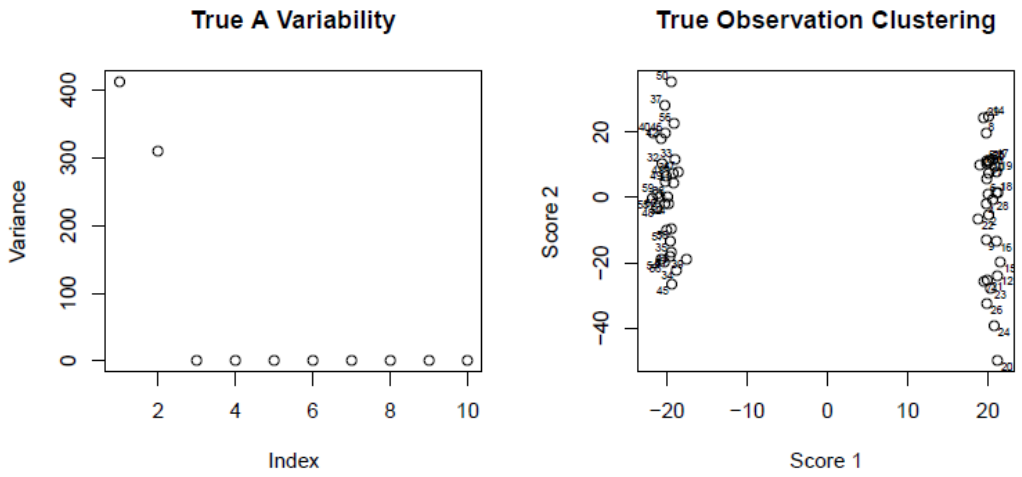

Figure 4.13: Variability of true A and true observational clustering (Simulation 2)

We set $k = 10$, as is the true intrinsic dimensionality, during the estimation. From the left panel of Figure 4.14, we can see the two most dominant layers are captured by our algorithms. And from the right, the clustering information is correctly reflected by the first column of the estimated $\mathbf{A}$. The corresponding dominant columns of the estimated loading matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ are displayed in Figure 4.15 and 4.16, respectively. By comparing the first two dominant loadings of $\hat{\mathbf{B}}$ to the dominant loadings of true $\mathbf{B}$, we conclude the true signals are reflected by our estimation in terms of both locations and magnitude and here the estimation suffers much less from noises compared to the estimation in the previous section. For the estimation of the loadings of $\mathbf{C}$, the true signals can be recovered almost perfectly and the non-signal portion is shrunk to zero clean and neat.

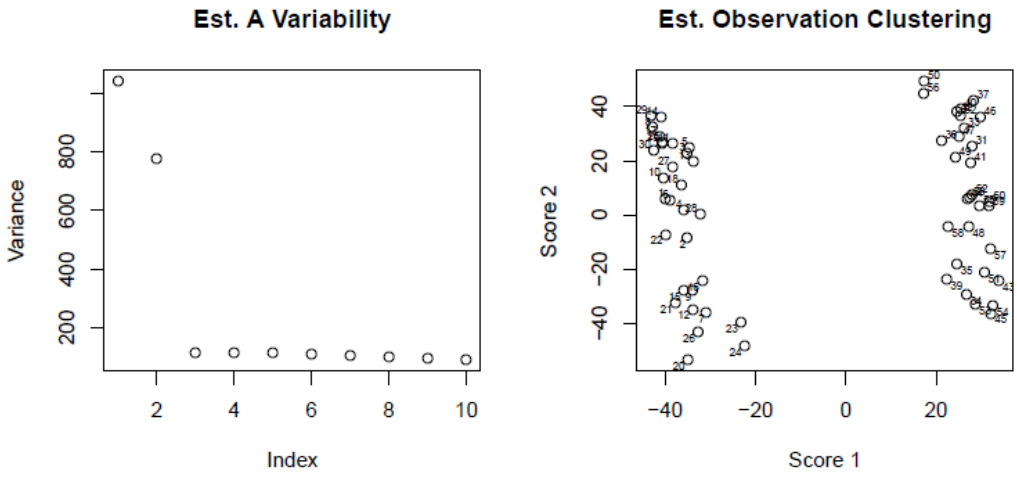

Figure 4.14: Variability of estimated A and estimated observational clustering (Simulation 2)

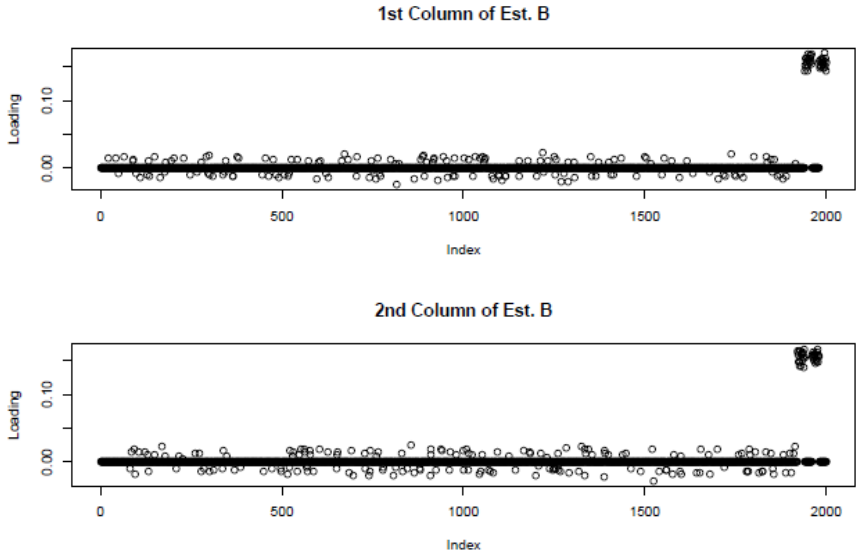

Figure 4.15: The first two columns of estimated B (Simulation 2)

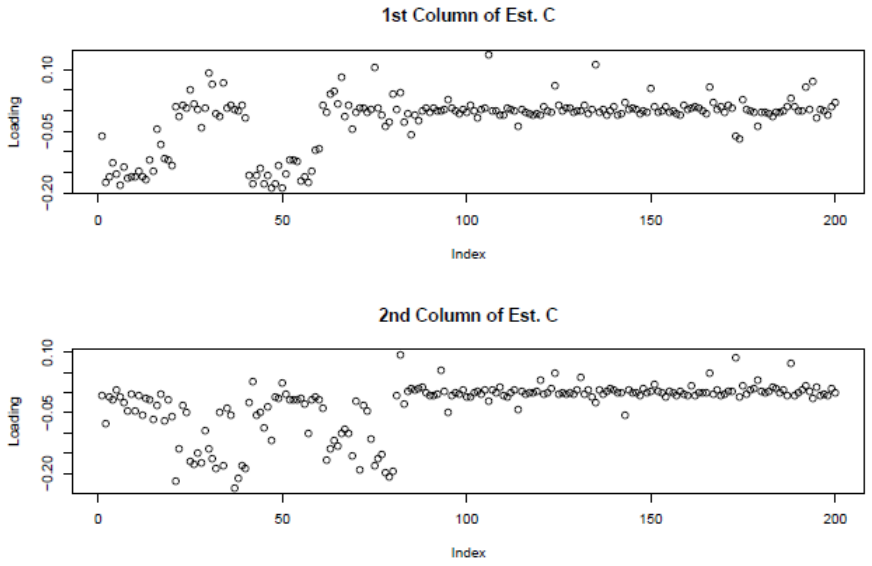

Figure 4.16: The first two columns of estimated C (Simulation 2)

As contrast, we run rCCA and PMD on these data sets. Figure 4.17 and 4.18 depict the leading loadings estimated by rCCA. Seen from the figures, the locations of the loadings for $\mathbf{X}$ can be captured correctly in spite of some inaccuracy in the magnitude and noises. While the estimation of the loadings for $\mathbf{Y}$ retrieves all the signals with limited noises regardless of magnitude. And its clustering information is in Figure 4.19.

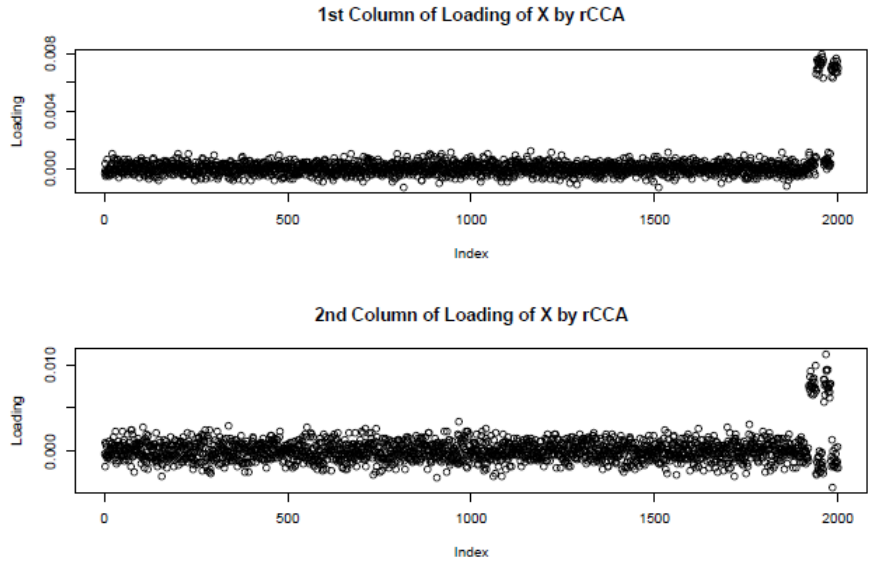

Figure 4.17: The first two columns of loadings of X by rCCA (Simulation 2)

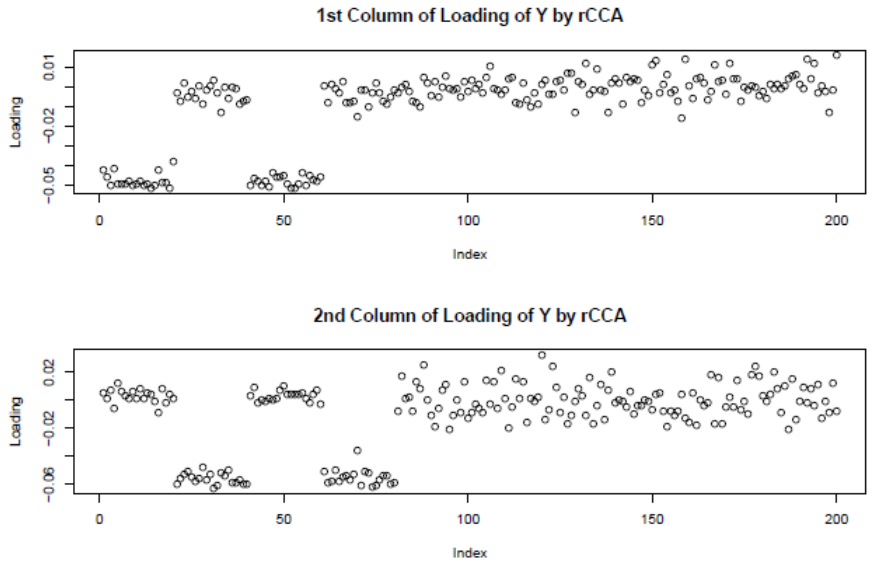

Figure 4.18: The first two columns of loadings of Y by rCCA (Simulation 2)

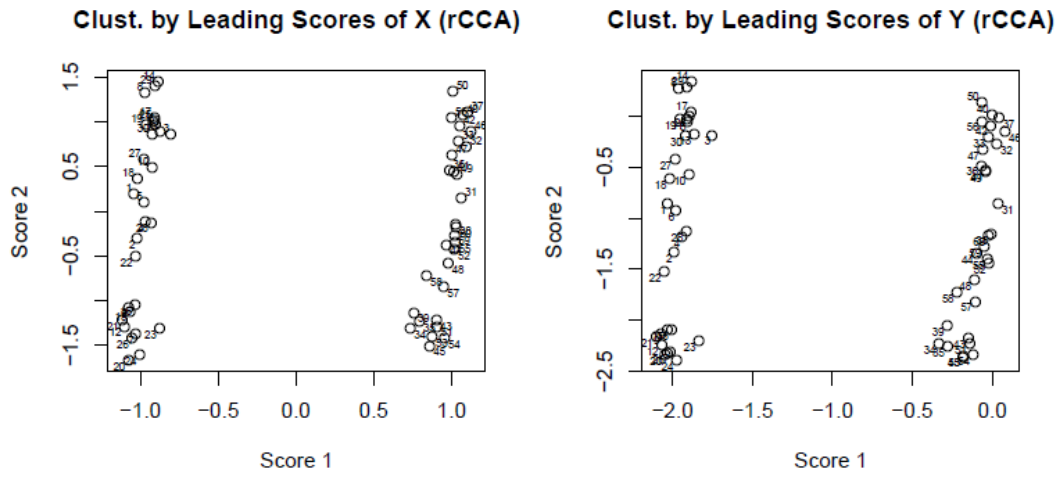

Figure 4.19: Clustering by leading scores of X and Y by rCCA (Simulation 2)

Applying PMD with optimally chosen tuning parameters to the data sets, Figure 4.20 and 4.21 show the first two estimated loadings of $\mathbf{X}$ and $\mathbf{Y}$. Seen from the plots, the loadings of $\mathbf{X}$ are extremely heavily penalized resulting in almost complete information loss; the estimated loadings of $\mathbf{Y}$ capture the signal locations of the truth with accurate estimates of the magnitude. As a cure, we consider PMD without any penalty at all with the hope of preserving the signals contained in $\mathbf{X}$. The estimated loadings with zero penalty by PMD are shown in Figure 4.22 and 4.23. With some noisy fluctuations, the signals contained in $\mathbf{X}$ can be captured, in this manner, with correct locations and approximate magnitude. And the estimation of the loadings for $\mathbf{Y}$ changes little compared to its optimally tuned counterpart. The estimated observation clustering is shown in Figure 4.24 with clear and consistent separation from either scores plot.

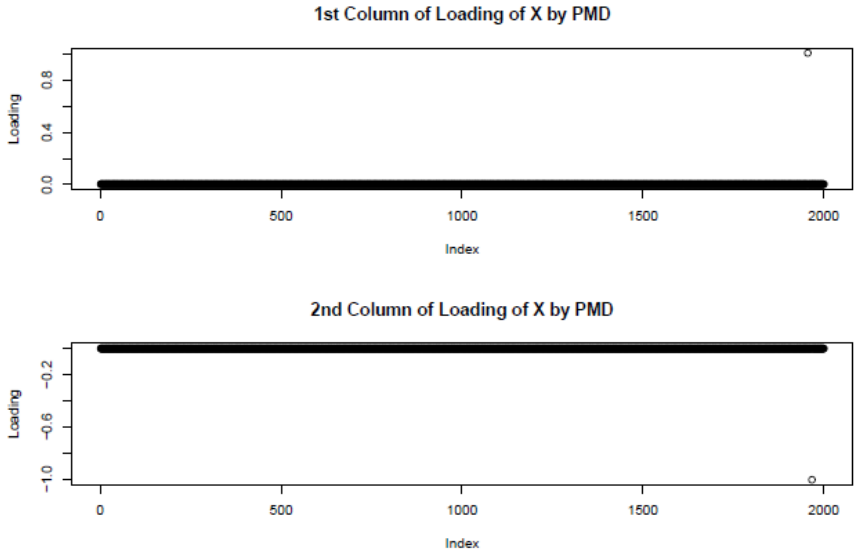

Figure 4.20: The first two columns of loadings of X by PMD (Simulation 2)

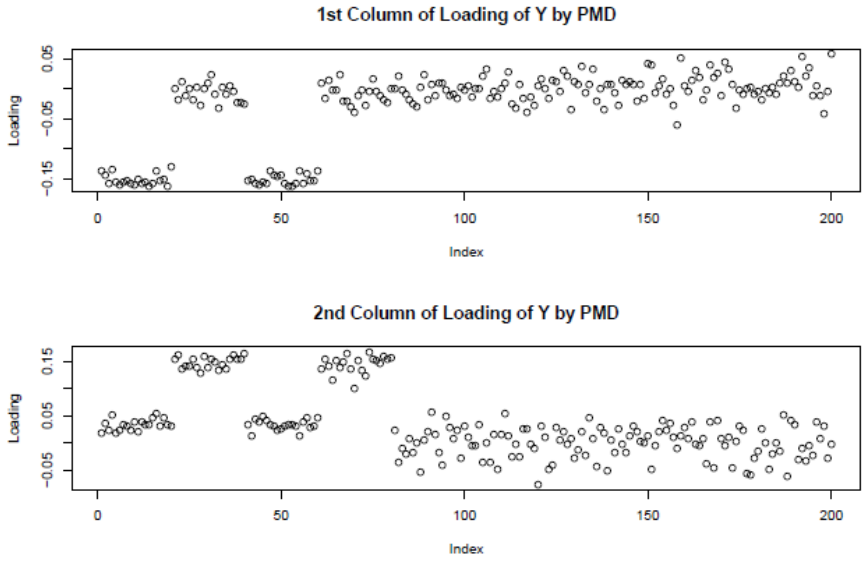

Figure 4.21: The first two columns of loadings of Y by PMD (Simulation 2)

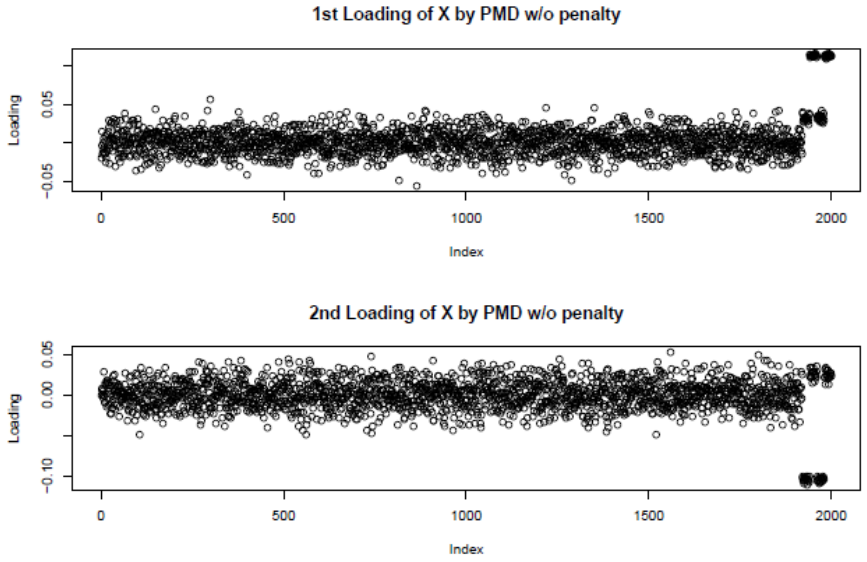

Figure 4.22: The first two columns of loadings of X by PMD w/o penalty (Simulation 2)

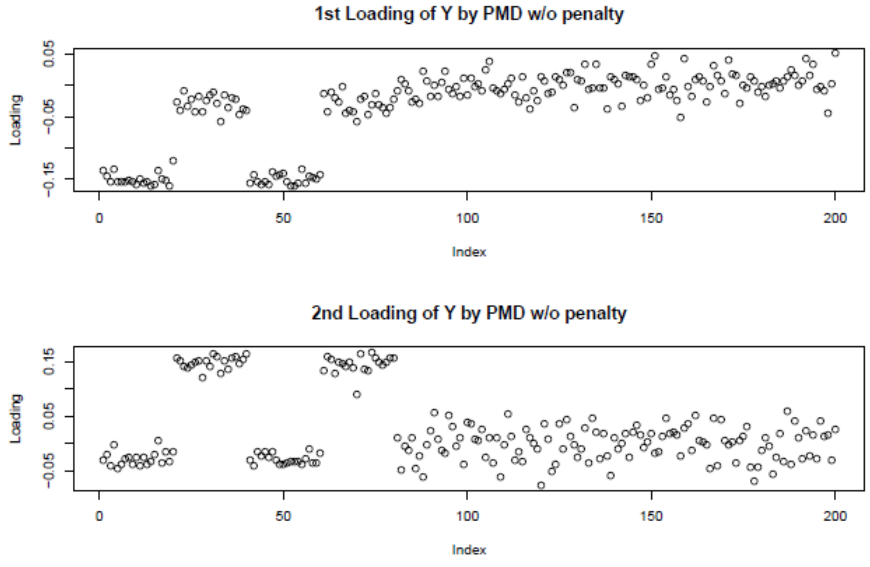

Figure 4.23: The first two columns of loadings of Y by PMD w/o penalty (Simulation 2)

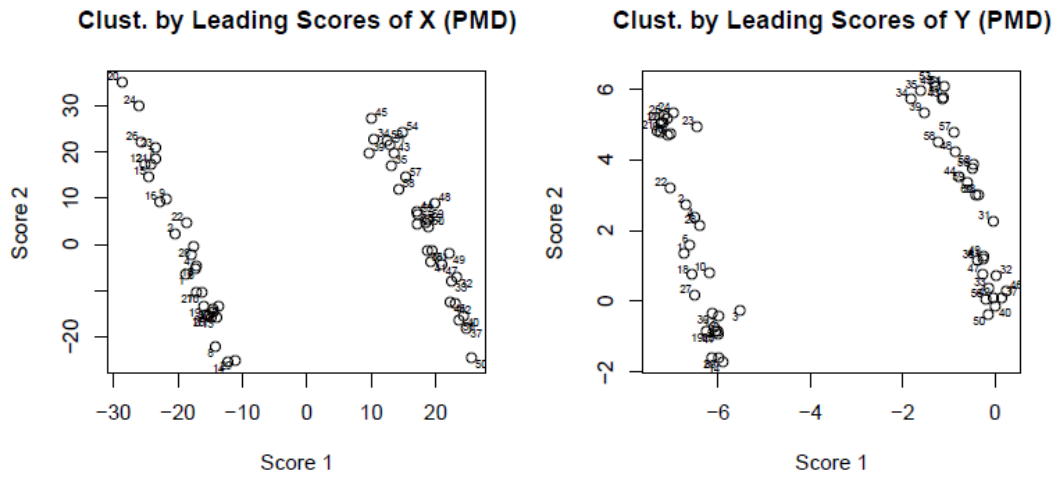

Figure 4.24: Clustering by leading scores of X and Y by PMD w/o penalty (Simulation 2)

Quantitatively, calculated as the average of 50 replications, the principal angle between the space spanned by the first two columns of $\hat{\mathbf{B}}$ and the space spanned by the corresponding columns of true $\mathbf{B}$ is 12.52° (with standard deviation 1.20°); the principal angle between the space spanned by the first two columns of $\hat{\mathbf{C}}$ and the column space of the first two columns of $\mathbf{C}$ is 29.14° (3.34°) by our approach. For rCCA, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is 43.68° (3.98°); the principal angle between the space spanned by the first two loadings of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is 20.08° (1.47°). By PMD with optimally chosen tuning parameters, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is 80.04° (6.12°); the principal angle between the space spanned by the first two loadings of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is 16.61°

(17.50°), while the un-penalized version produces 46.96° (1.40°) and 19.43° (1.42°), respectively.

4.1.3 Study with Large Dimensionality and Small Sample Size (Simulation 3).

In this simulation study, we put our method to the test in a much more challenging setting by increasing the number of genes by a factor of 5 and the number of SNPs by a factor of 10, which is very close to the scale of a typical GWAS. We again generate the data sets from the model, specifically, by equation (2.1) and (2.3). Set $n = 30$ (small sample size), the true intrinsic dimension of both data sets $k = 10$, $d_1 = 10000$ and $d_2 = 2000$. Matrix $\mathbf{A}$ is generated column-wise exactly as how it is done in simulation 1. The variances of the columns of the true $\mathbf{A}$ generated are plotted in the left panel of Figure 4.25, and we should expect two layers of dominant signals based on the plot and the clustering information is contained in the right. For the generation of true $\mathbf{B}_{10000\times 10}$, the first column sequentially has 9700 0's, 100 -1's, 100 0's and 100 -1's; the second column has 9600 0's, 100 1's, 100 0's, 100 1's and 100 0's; for column 3 to column 10, 10% of the entries are set at random to 1 and the rest are set to 0. And then each column of the generated true $\mathbf{B}$ is re-scaled to have length unity. Figure 4.26 depicts the first two columns of the true $\mathbf{B}$ generated. The true $\mathbf{C}_{2000\times 10}$ is determined as the following: The first column is sequentially comprised of 200 1's, 200 0's, 200 1's and 1400 0's; the second column has 200 0's, 200 -1's, 200 0's, 200 -1's and 1200 0's; for column 3 to column 10, 10% of the entries are randomly set to 1 and the rest are set to 0. And then each column of the generated true $\mathbf{C}$ is re-scaled to have length unity. The plots of the first two columns of the true $\mathbf{C}$ can be found in Figure 4.27. Besides, μ and ν are set to be zero vectors, and σ^2, the variance of the errors, is set to equal 1. Then, matrix $\mathbf{X}$ can be formed by equation (2.1) with μ, $\mathbf{A}$, $\mathbf{B}$ and errors generated. And

matrix $\mathbf{Y}$ is generated as follows: Form matrix $\mathbf{\Theta}$ by equation (2.3) with ν, $\mathbf{A}$ and $\mathbf{C}$ given and y_{ij} is generated from a Bernoulli with success probability $\pi_{ij} = \pi(\theta_{ij})$ independently.

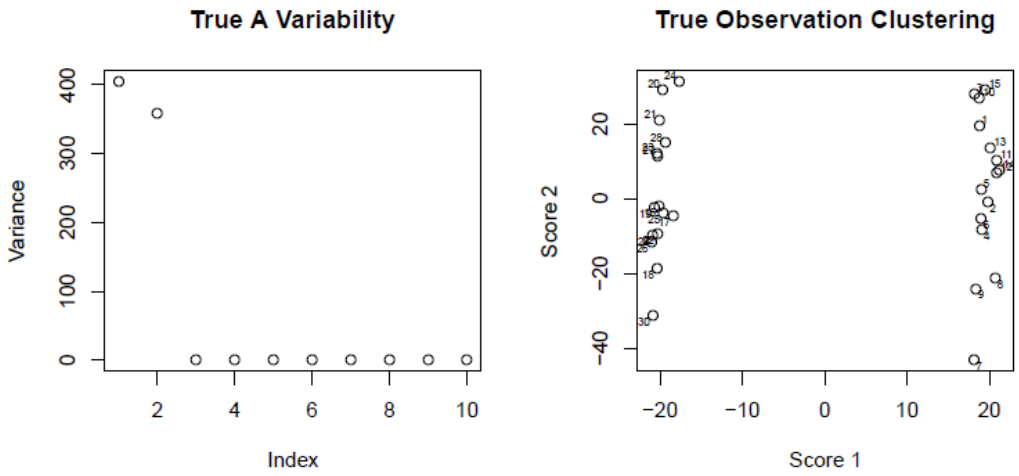

Figure 4.25: Variability of true A and true observational clustering (Simulation 3)

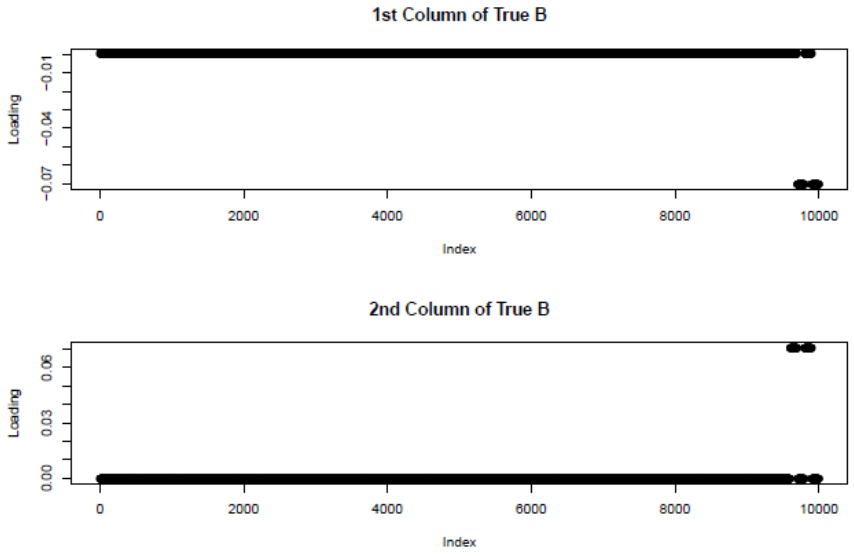

Figure 4.26: The first two columns of true B (Simulation 3)

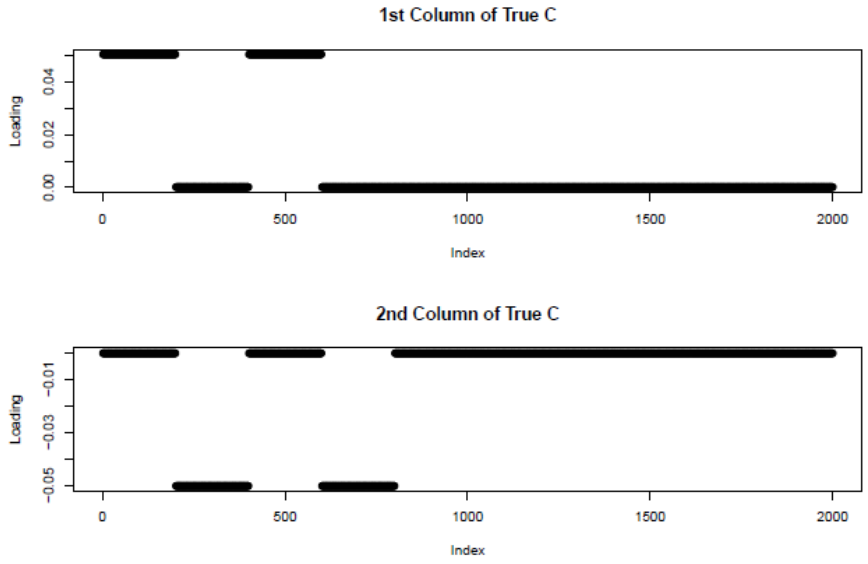

Figure 4.27: The first two columns of true C (Simulation 3)

Set $k = 10$ as usual during the estimation process. The two most dominant layers stand out in terms of variability as depicted by the left panel of Figure 4.28. The corresponding columns of the estimated loading matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ are displayed in Figure 4.29 and 4.30, respectively. Comparing the plots in Figure 4.29 to the truth, it is safe to conclude all the signals in $\mathbf{X}$ can be recovered and they stand out from the noises quite clean, especially given this high dimensionality. However, the plots of the estimated $\mathbf{C}$ suffer greatly from noises, a sign of under-penalization. But we can still see a great proportion of the signals are distinct when compared to the background noises.

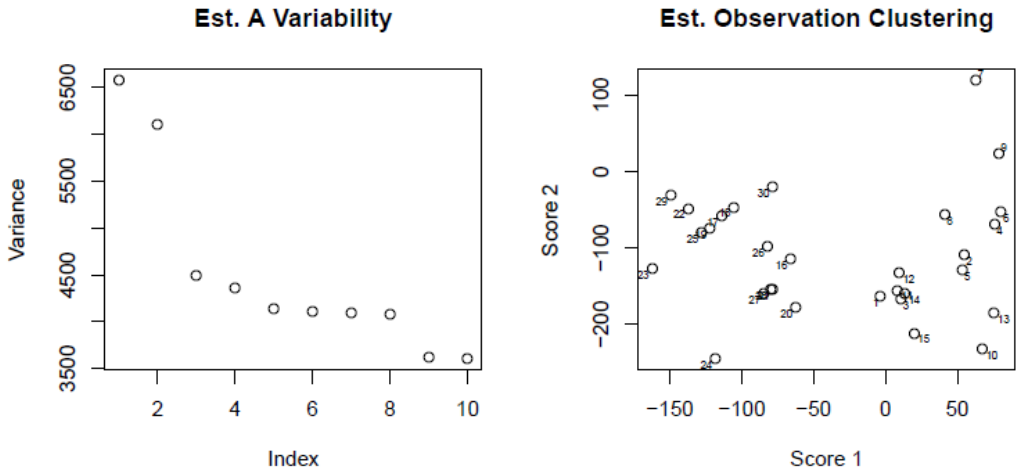

Figure 4.28: Variability of estimated A and estimated observational clustering (Simulation 3)

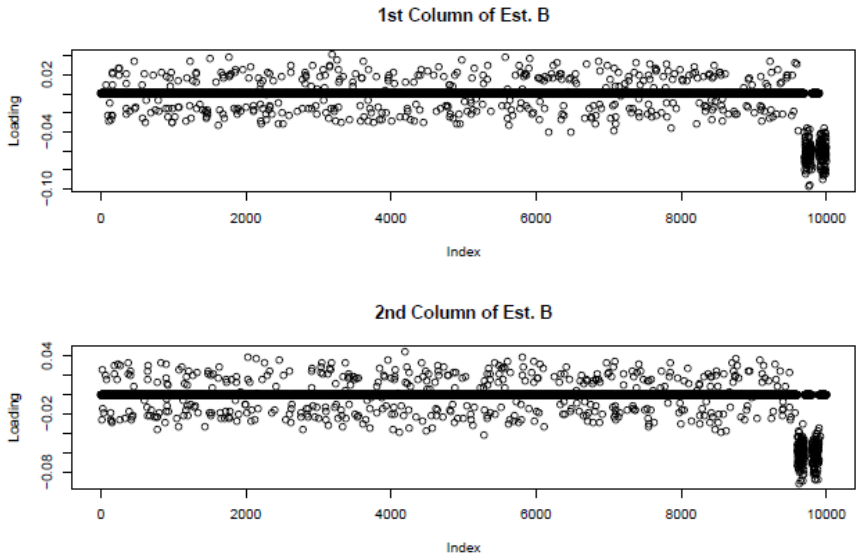

Figure 4.29: The first two columns of estimated B (Simulation 3)

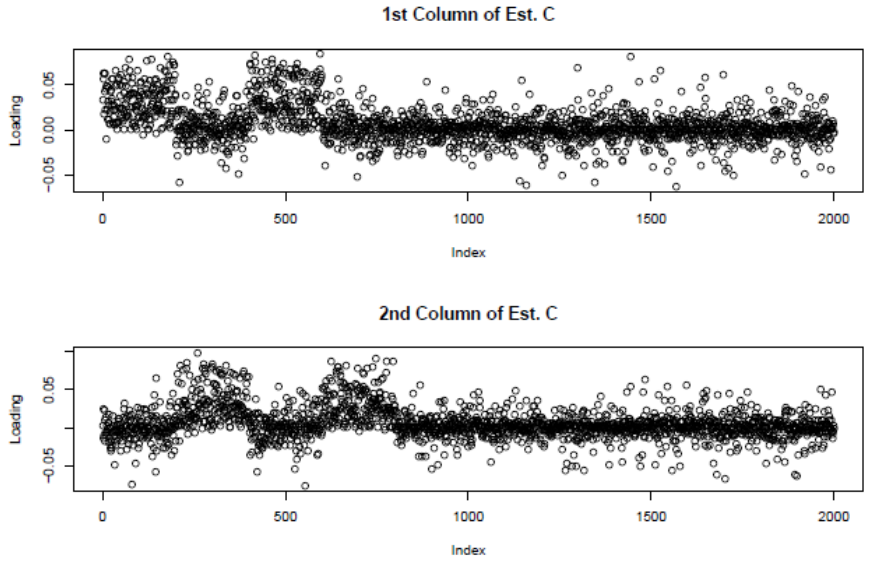

Figure 4.30: The first two columns of estimated C (Simulation 3)

For the competing approaches, rCCA no longer works as the calculation of $\mathbf{X}^T\mathbf{X}$, required by the rCCA algorithm, risks depleting the memory for such a large data set. Applying PMD to the data sets, Figure 4.31 and 4.32 show the first two estimated loadings of $\mathbf{X}$ and $\mathbf{Y}$. Seen from the plots, the loadings of $\mathbf{X}$ seem over-penalized and a non-negligible proportion of signals are shrunk to zero. But the loadings of $\mathbf{Y}$ are estimated satisfactorily by reflecting the true locations and magnitude of the signals with only moderate noise level. For the record, the estimated clustering information can be found in Figure 4.33.

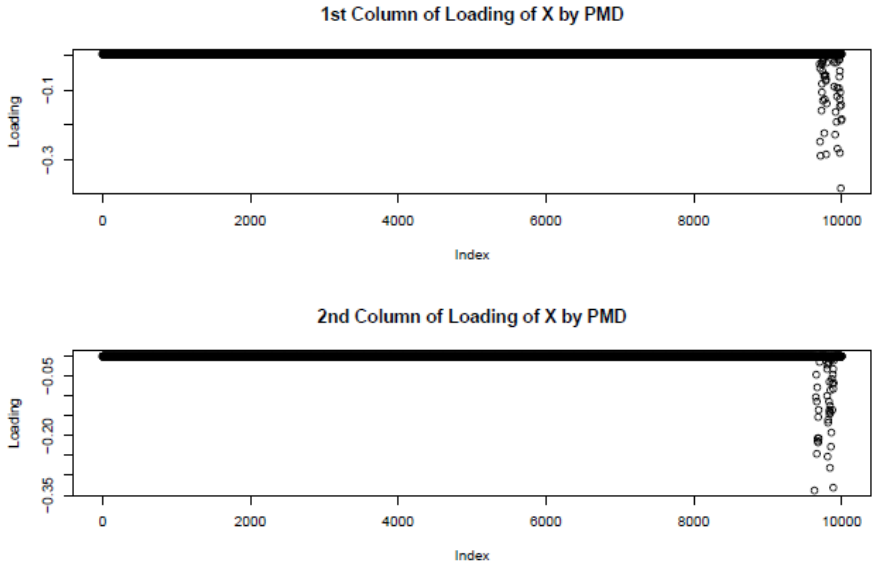

Figure 4.31: The first two columns of loadings of X by PMD (Simulation 3)

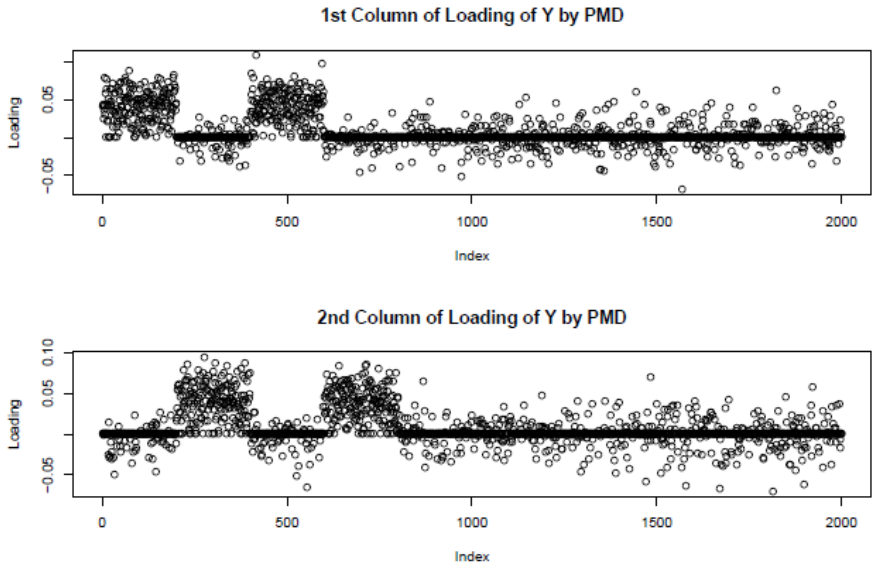

Figure 4.32: The first two columns of loadings of Y by PMD (Simulation 3)

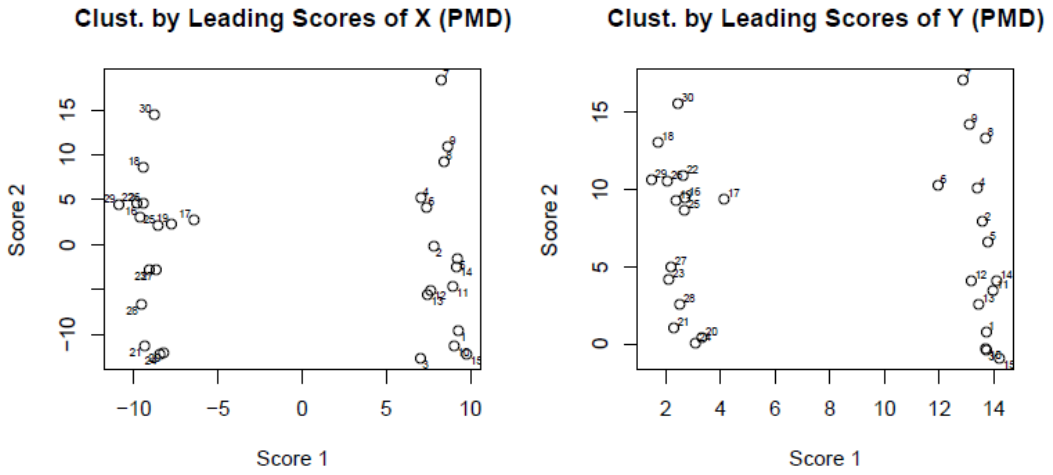

Figure 4.33: Clustering by leading scores of X and Y by PMD (Simulation 3)

Supplementarily, with 20 replications, the average principal angle between the

space spanned by the first two columns of $\hat{\mathbf{B}}$ and the space spanned by the corresponding columns of true $\mathbf{B}$ is 33.82° (5.30°); the principal angle between the space spanned by the first two columns of $\hat{\mathbf{C}}$ and the column space of the first two columns of $\mathbf{C}$ is 56.12° (3.98°) by our approach. With optimally tuned PMD, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is on average 85.08° (3.97°); the principal angle between the space spanned by the first two loadings of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is on average 50.20° (4.27°).

4.1.4 Study with Large Dimensionality and Large Sample Size (Simulation 4).

The simulation study in this section is a reenactment of Simulation 3 with sample size increased to $n = 60$ and everything else kept the same. Specifically, we set $k = 10$, $d_1 = 10000$ and $d_2 = 2000$. Matrix $\mathbf{A}$ is generated column-wise the same as how it is in Simulation 3 except the first column assigns the first 30 subjects to a group and the rest 30 to the other. The variances of the columns of the true $\mathbf{A}$ generated are plotted in the left panel of Figure 4.34 and the clustering information is contained in the right panel. The loadings $\mathbf{B}$ and $\mathbf{C}$ are set the same as they are in Simulation 3 and the plots of their leading columns can be found in Figure 4.26 and 4.27 respectively.

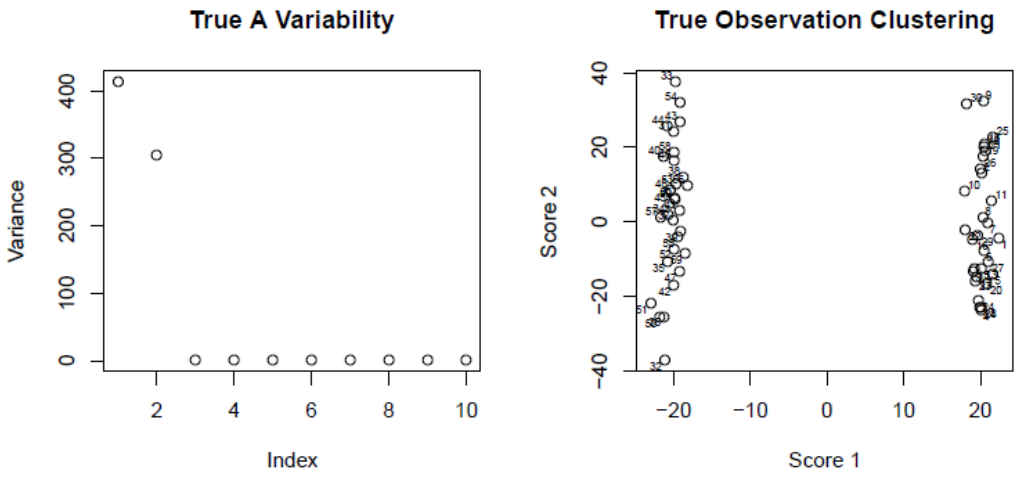

Figure 4.34: Variability of true A and true observational clustering (Simulation 4)

We set $k = 10$ during the estimation. The left panel of Figure 4.35 indicates the two most dominant layers are captured by our algorithms. If we take a closer look at the variability of the estimated columns and compare to the same plot in the previous section, we can see the increased sample size plays a significant role in reducing the estimated variability and stabilizing the estimation. And this can also be reflected by the right panel of Figure 4.35, where we have a clearer separation of the observations compared to the right panel of Figure 4.28. The corresponding dominant columns of the estimated loading matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ are displayed in Figure 4.36 and 4.37, respectively. As shown by Figure 4.36, our algorithm perfectly recovers all the signals and shrinks all the noises to zero. As for the estimated loadings in $\mathbf{C}$, Figure 4.37 suggests much less noises and clearer standing-out of the signals compared to the corresponding estimation in the previous section, although still under-penalized, illustrating sample size effect in helping come up with better estimation, just as expected.

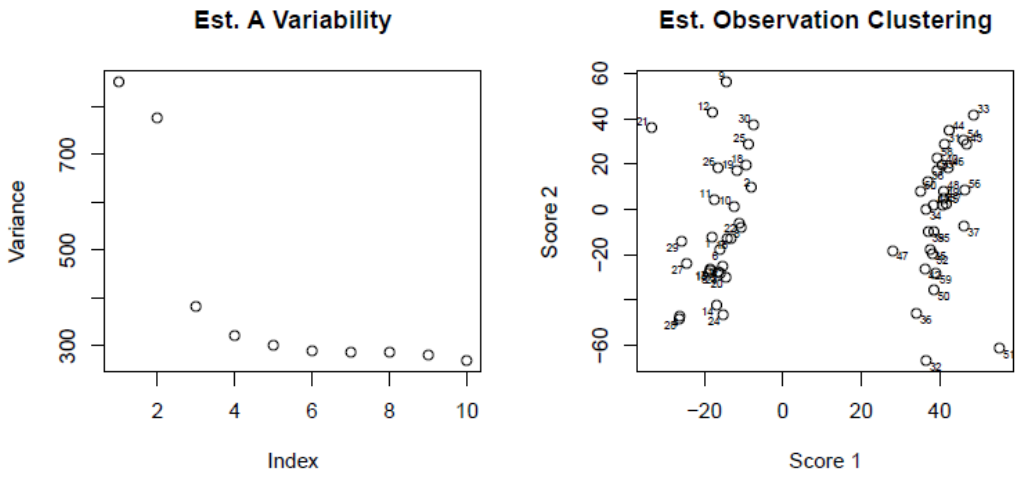

Figure 4.35: Variability of estimated A and estimated observational clustering (Simulation 4)

Figure 4.36: The first two columns of estimated B (Simulation 4)

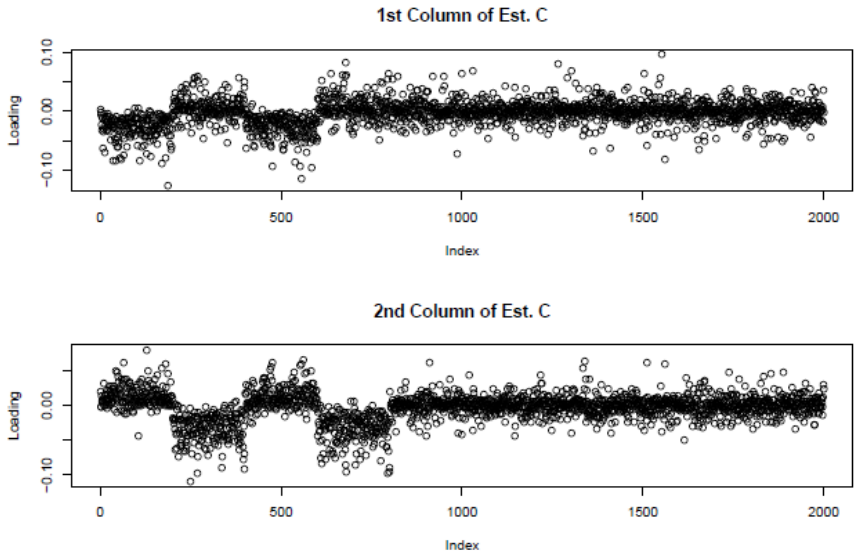

Figure 4.37: The first two columns of estimated C (Simulation 4)

Applying optimally tuned PMD to the data sets, Figure 4.38 and 4.39 show the first two estimated loadings of $\mathbf{X}$ and $\mathbf{Y}$. Seen from the plots, the loadings of $\mathbf{X}$ are again extremely heavily penalized resulting in almost complete information loss; the estimated loadings of $\mathbf{Y}$ are able to capture the signal locations of the truth with reasonably accurate estimates of the magnitude in general but still suffer from under-penalization indicated by noticeable fluctuations. As the optimally tuned PMD fails to capture the signals in $\mathbf{X}$, we supplement the analysis with PMD without any penalty at all, expecting the missed signals can be preserved. The estimated loadings with zero penalty by PMD are shown in Figure 4.40 and 4.41. As expected, the signals in the continuous data set can be captured although with great noises in this manner. And zero penalty does not change very much the estimated loadings of the binary data set compared to the optimally tuned version. However, a closer look indicates a slightly worse off first layer and a better off second layer compared

to the optimally chosen version. The estimated observation clustering is shown in Figure 4.42 with clear and consistent separation from either scores plot.

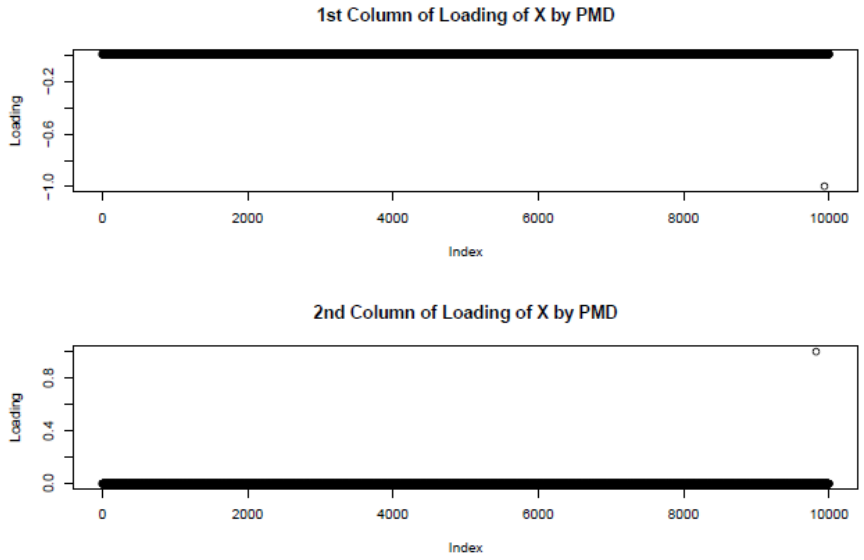

Figure 4.38: The first two columns of loadings of X by PMD (Simulation 4)

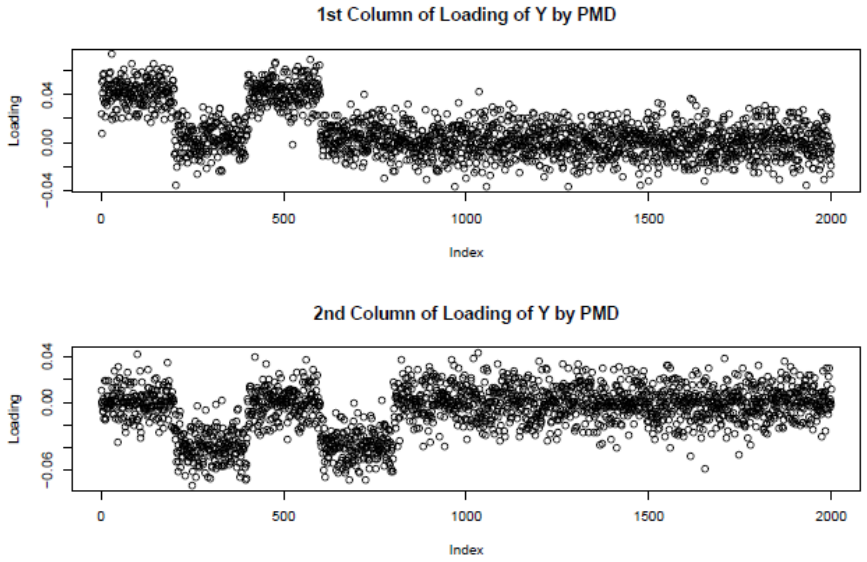

Figure 4.39: The first two columns of loadings of Y by PMD (Simulation 4)

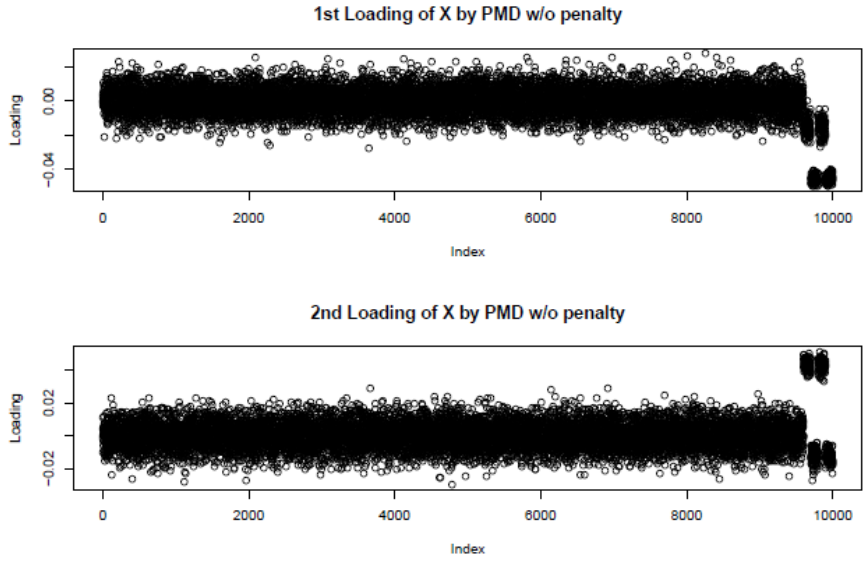

Figure 4.40: The first two columns of loadings of X by PMD w/o penalty (Simulation 4)

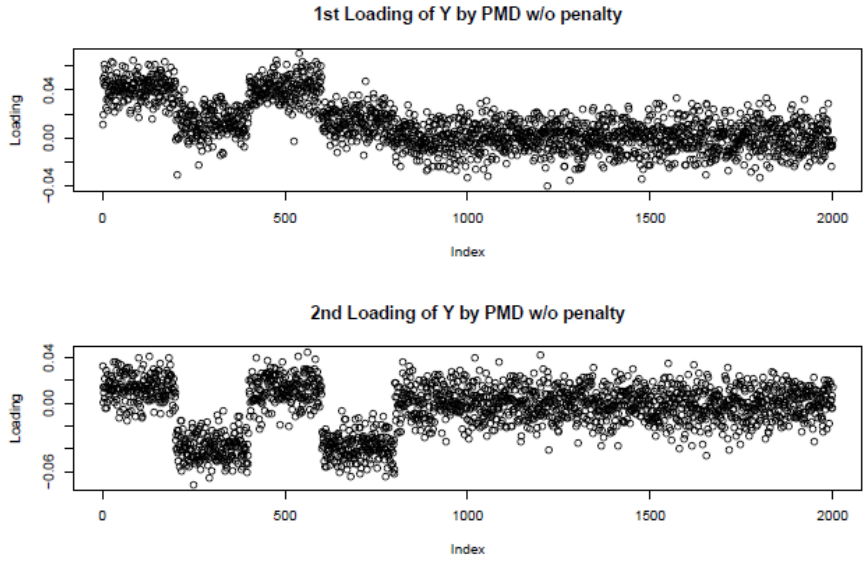

Figure 4.41: The first two columns of loadings of Y by PMD w/o penalty (Simulation 4)

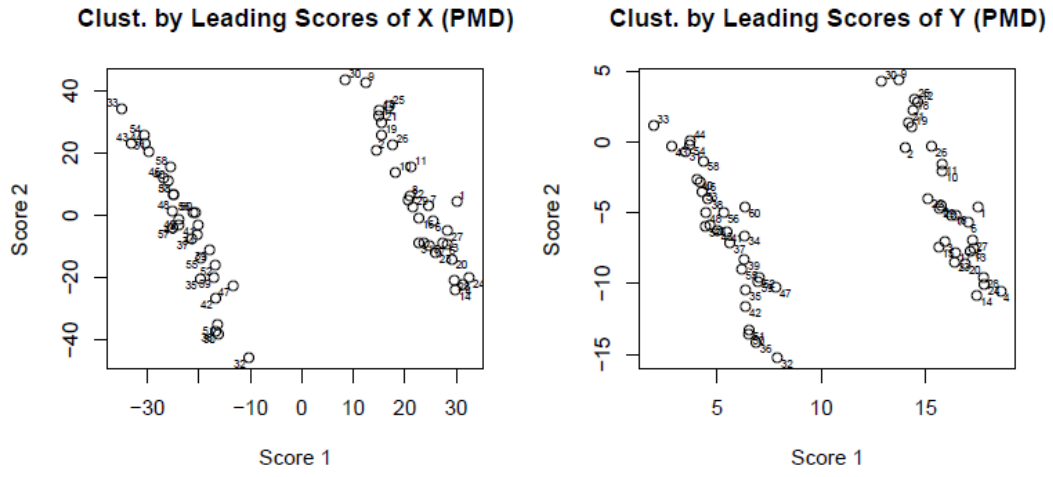

Figure 4.42: Clustering by leading scores of X and Y by PMD w/o penalty (Simulation 4)

With a quantitative measure based on 20 replications, the principal angle between the space spanned by the first two columns of $\hat{\mathbf{B}}$ and the space spanned by the corresponding columns of true $\mathbf{B}$ is 19.95° (14.87°); the principal angle between the space spanned by the first two columns of $\hat{\mathbf{C}}$ and the column space of the first two columns of $\mathbf{C}$ is 57.33° (4.37°) by our approach. By PMD with optimally chosen tuning parameters, the principal angle between the space spanned by the first two loadings of $\mathbf{X}$ and the space spanned by the first two columns of true $\mathbf{B}$ is 85.95° (6.20×10^{-7}°); the principal angle between the space spanned by the first two loadings of $\mathbf{Y}$ and the space spanned by the first two columns of true $\mathbf{C}$ is 40.40° (2.80°), while the un-penalized version produces 50.85° (1.16°) and 37.26° (1.86°), respectively.

As a conclusion based on the simulation studies with various setups in this section, rCCA and PMD may capture some hidden information to some extent although they are not designed purposefully for this kind of data analysis, putting aside the fact that rCCA does not work on large data sets for memory issues and PMD tends to impose very heavy penalties leading to significant amount of signal loss for the continuous data set and the tuning parameter selection is unstable as randomness plays a very noticeable role in it. As contrast, our method outperforms rCCA and PMD in general by accurately recovering the shared information of systematical connection embedded within a continuous and a binary data set, offering natural interpretations of a layer-by-layer subnetwork-to-subnetwork association structure where an order of importance of the layers can be established systematically by rationale similar to singular value decomposition or principal component analysis as demonstrated by our extensive simulation studies with various setups.

4.2 Simulation with Data Generated Not from the Model (Simulation 5)

Complementarily, we consider two other data sets with correlation structures incorporated by a regression setup and we expect to see those incorporated hidden structures are somehow revealed by our method.

4.2.1 Simulation Setup

Recall the dimensions of the continuous matrix $\mathbf{X}_{n\times d_1}$ and the binary matrix $\mathbf{Y}_{n\times d_2}$. In this study, we arbitrarily set $n = 100$, $d_1 = 10$ and $d_2 = 20$. First, we generate the binary data matrix $\mathbf{Y}$. Consider the submatrix formed by the first 50 rows and first 10 columns of $\mathbf{Y}$: for each individual column of that submatrix, set randomly 80% of the entries to 1 and the rest to 0, and for the rest of $\mathbf{Y}$ rather than the submatrix above, set at random 10% of the entries to 1 and rest to 0. The purpose of this setting is to concentrate the signals to the first 10 variables of $\mathbf{Y}$. And then we generate the continuous matrix $\mathbf{X}$ formed by a multi-task regression relationship $\mathbf{X} = \mathbf{Y}\mathbf{S}_{lp} + \mathbf{E}$, where $\mathbf{E}$ is a matrix of standard normal random errors and $\mathbf{S}_{lp_{20\times 10}}$ is a matrix of slopes and is defined as the following: set the first 5 entries of the first column of $\mathbf{S}_{lp}$ to 50, the first 5 entries of the second column to -40 and the first 5 entries of the third column to 30, and the rest entries in $\mathbf{S}_{lp}$ are generated uniformly from -0.5 to 0.5. As a consequence to this setting, we expect to see the signals in $\mathbf{X}$ concentrate on the first 3 columns. Moreover, we should also expect to distinguish the first 50 observations from the rest 50. The heat maps of the generated $\mathbf{X}$ and $\mathbf{Y}$ are shown in Figure 4.43 and 4.44, respectively. In the heat maps, in particular, a lighter color indicates a larger value. The light bars at the bottom of the first three columns in Figure 4.43 contain the signals in $\mathbf{X}$, while the bottom left corner of Figure 4.44 indicates where the signals lie in $\mathbf{Y}$. Moreover, the first 50 observations can be distinguished from the rest 50, seen from either heat

map.

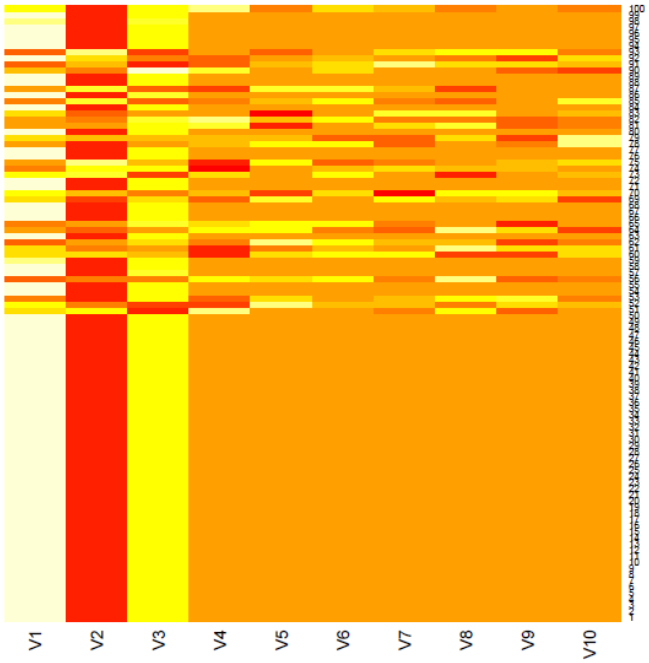

Figure 4.43: Heat map of X

4.2.2 Simulation Results

We set $k = 10$ in the algorithms during the estimation process, as 10 is the maximum value allowed for k before singularity emerges. In the left portion of Figure 4.45, the variability of the estimated score matrix is plotted and we conclude there is only one dominant layer of signals. Then, the first column of the estimated loading matrices are shown in Figure 4.46. As expected, the first 10 variables of $\mathbf{Y}$ and the first 3 variables of $\mathbf{X}$ are picked up as the magnitude of the corresponding values in the loadings deviates the most from zero. A closer look at the first panel of Figure 4.46 suggests the signs of the loadings of $\mathbf{X}$ are in accordance with the

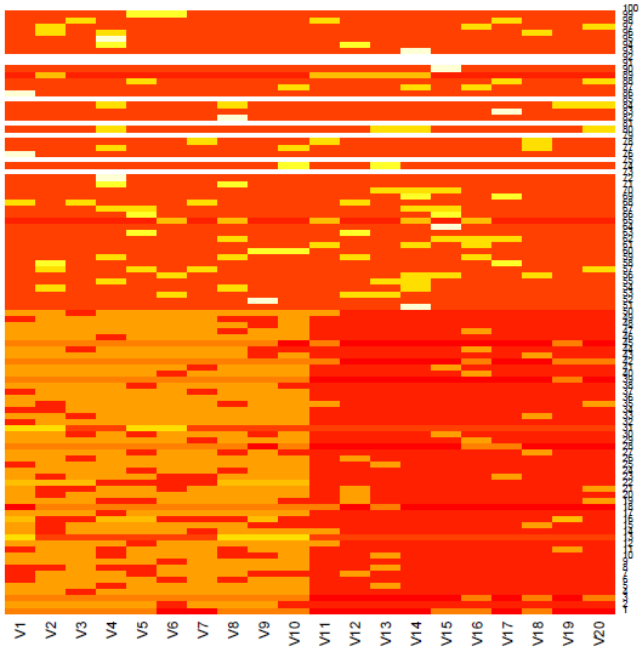

Figure 4.44: Heat map of Y

signs of the top five entries of the first three columns of $\mathbf{S}_{lp}$ up to a flip. Apart from that, the two clusters of observations are revealed in the right panel of Figure 4.45 when plotting the first score against the indices of the observations. Hence, all the hidden structures embedded in the data sets by multi-task regression relationships are accurately reflected by our approach.

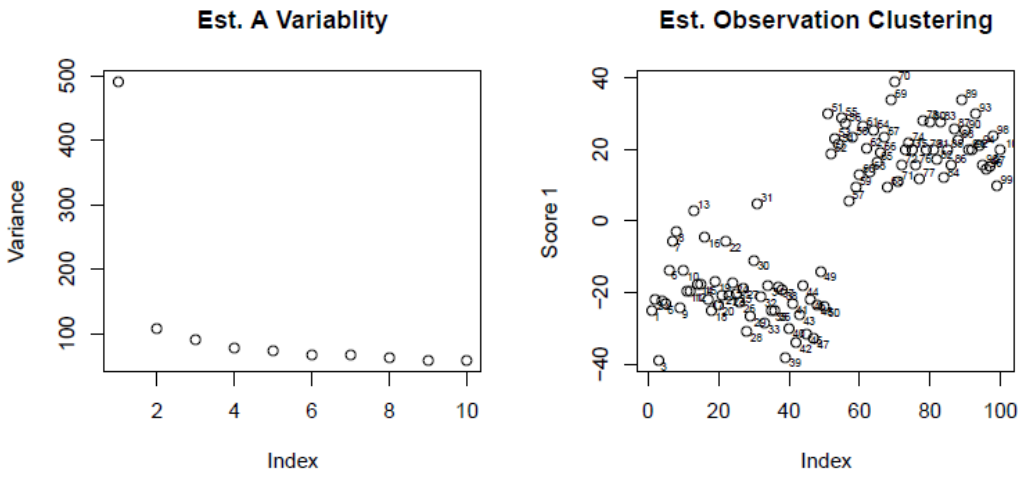

Figure 4.45: Variability of estimated A and estimated observational clustering (Simulation 5)

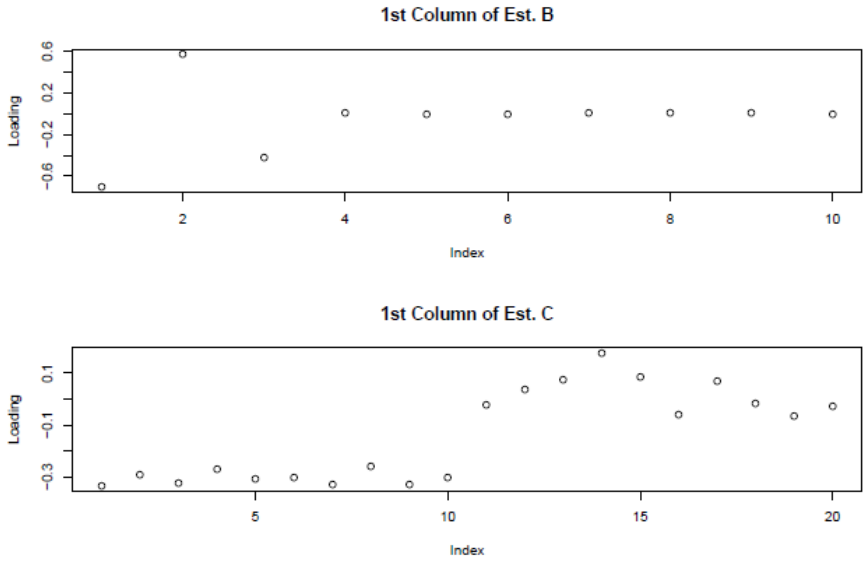

Figure 4.46: The first column of estimated B and estimated C (Simulation 5)

For carefully tuned rCCA, the first loading of $\mathbf{X}$ and $\mathbf{Y}$ can be found in Figure 4.47, from which we can see the estimated loading of the continuous data set is identical to our estimation regardless of scaling issue, but the estimated loading of the binary data set only captures the first 5 variables instead of 10. The estimated observational clustering is shown in Figure 4.48, consistent but not as clean as it is by our approach.

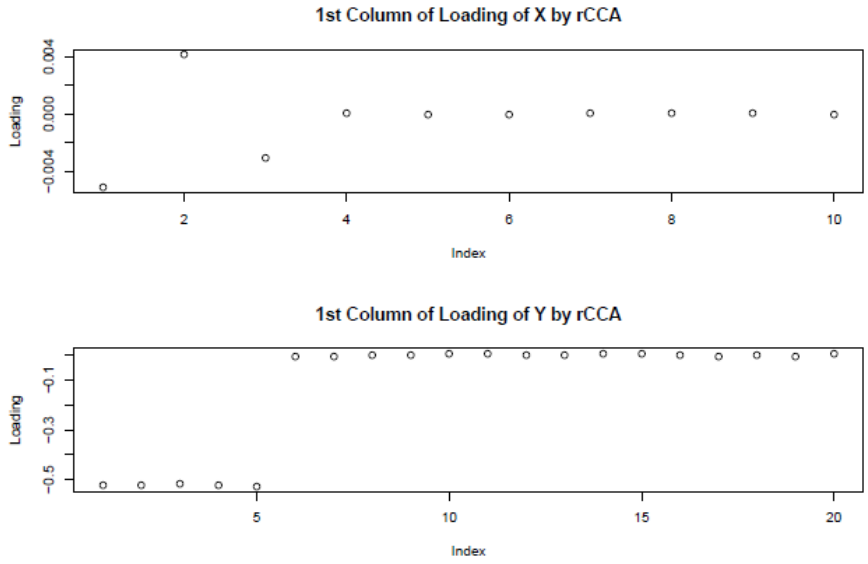

Figure 4.47: The estimated first loading of X and Y by rCCA (Simulation 5)

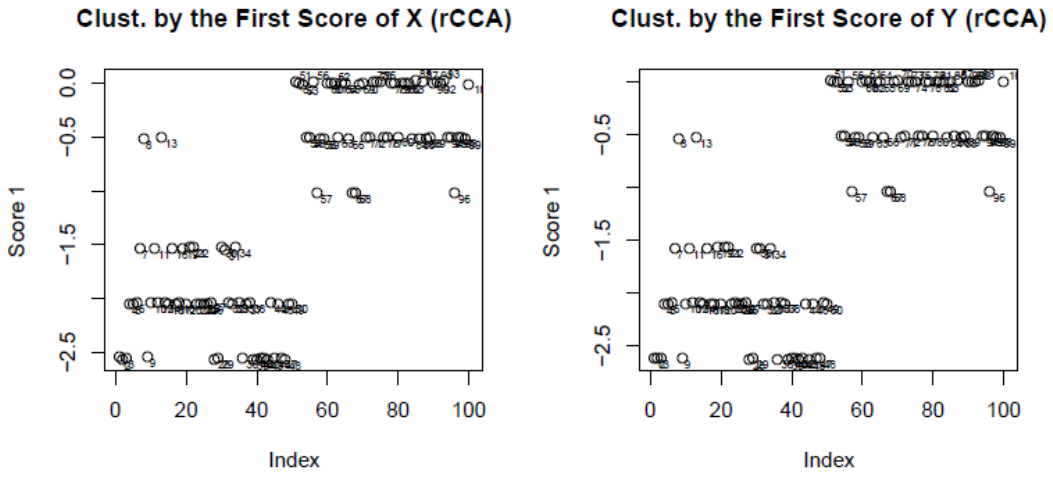

Figure 4.48: The estimated observational clustering by rCCA (Simulation 5)

As for PMD with optimally chosen tuning parameters, the first loading of $\mathbf{X}$ and $\mathbf{Y}$ are depicted in Figure 4.49, where the estimated loading of $\mathbf{X}$ misses the first two variables while the estimated loading of $\mathbf{Y}$ can be considered adequate as 9 out of the first 10 variables are retrieved. And the estimated observational clustering by PMD is shown in Figure 4.50.

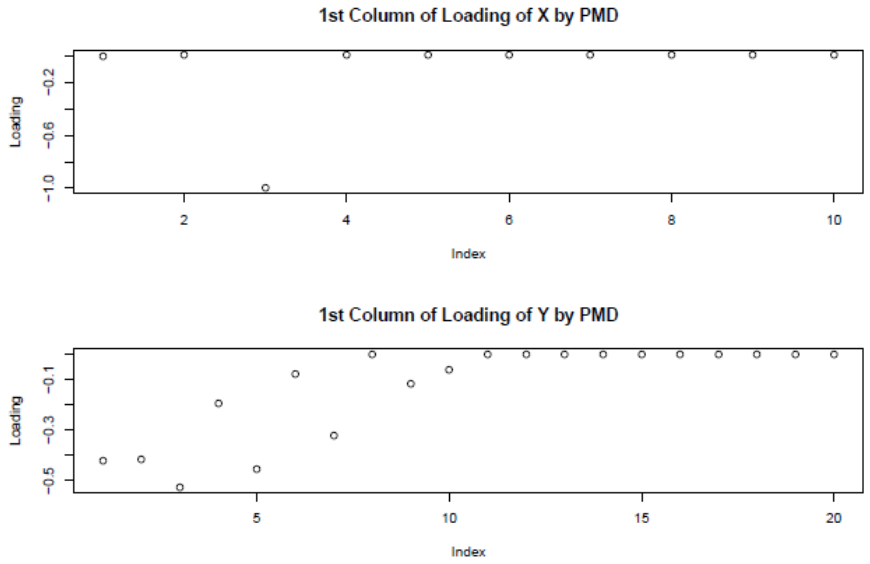

Figure 4.49: The estimated first loading of X and Y by PMD (Simulation 5)

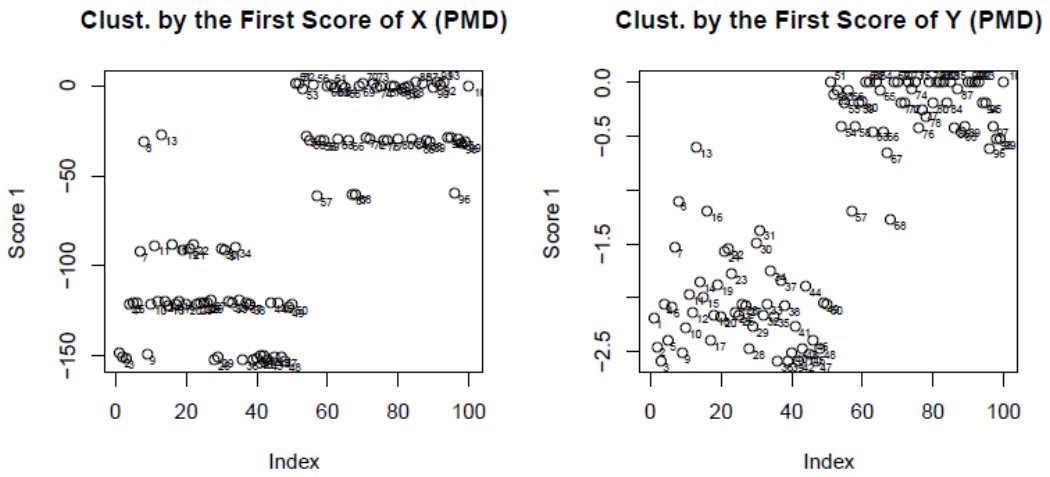

Figure 4.50: The estimated observational clustering by PMD (Simulation 5)

5. APPLICATION TO REAL DATA

With advances in biomedical sciences and technologies, such as DNA/RNA sequencing and microarray analysis, a genome-wide association study (GWA study or GWAS) has become a popular way to examine a large set of common genetic variants in a group of individuals of interest to detect any genetic variant associated with a trait. In general, GWAS focuses on identifying associations between SNPs and gene expression levels of many traits. For that purpose, eQTL mapping offers a way to search for significant associations between genetic variants and gene expression. Our proposed method, by design, can potentially be used for eQTL mapping with a natural interpretation that a set of selected genetic variants jointly associate with and co-regulate the expression of a set of selected genes. And another popular method, with favorable underlying subgroup-to-subgroup structure and applicable to the scale of the data analysis task in this section, the penalized matrix decomposition (worth mentioning that rCCA is not computationally feasible), is carried out as a comparison.

For the purpose of demonstration, we use the BXD gene expression data and the BXD marker data. The BXD gene expression data set is available from the website genome.unc.edu and is described in Gatti et al. (2007). Briefly, it consists of gene expression measurements for 20868 transcripts (each corresponding to the expression of a particular gene) in the liver by microarray in 39 BXD recombinant inbred strains and the C57BL/6J and DBA/2J parentals (mice). The BXD gene expression data set has 41 observations and is treated as the continuous data set $\mathbf{X}$. The observations are denoted by the following, BXD1, BXD2, BXD5, BXD6, BXD8, BXD9, BXD11, BXD12, BXD13, BXD14, BXD15, BXD16, BXD19, BXD21,

BXD23, BXD24, BXD28, BXD29, BXD31, BXD32, BXD33, BXD34, BXD38, BXD39, BXD40, BXD42, BXD43, BXD44, BXD45, BXD48, BXD51, BXD60, BXD62, BXD69, BXD73, BXD77, BXD85, BXD86, BXD92, C57BL/6J and DBA/2J. On the other hand, the BXD marker data are comprised of 3795 informative markers (each corresponding to an SNP in a particular chromosome), directly downloadable for free from http://www.genenetwork.org/genotypes/BXD.geno, with further information, detailed description about the experiment and sources of data available at http://www.genenetwork.org/dbdoc/BXDGeno.html. The marker data set, which is binary, has the same 41 observations and is treated as $\mathbf{Y}$.

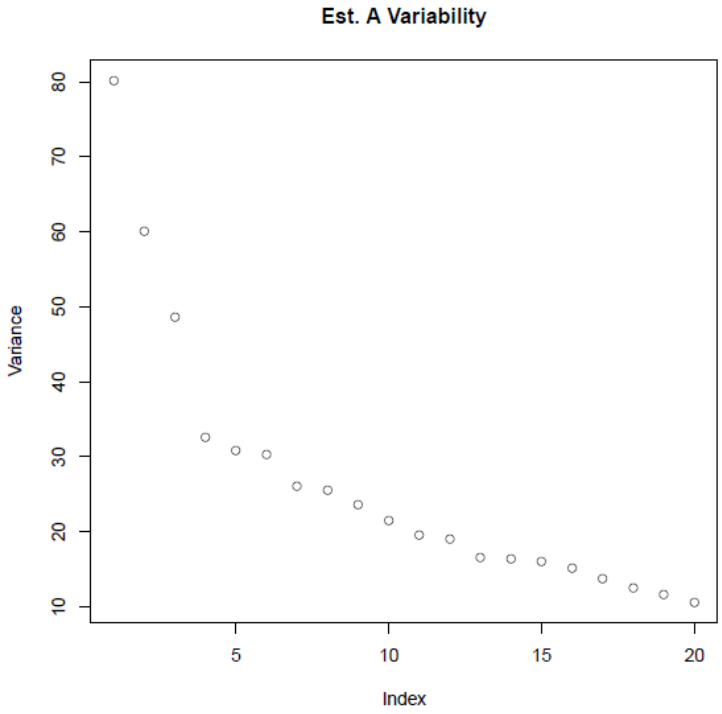

Figure 5.1: Variances of each column of estimated A (Real Data)

The analysis in this section tries to relate the dominant gene expression in liver to

a set of selected markers. The biomedical significance is justified by referring to genes with confirmed expression level in liver and co-regulation is depicted by visualizing gene pathways. We applied our algorithms described previously to the two data sets. The dimensionality was set to be 20, that is, $k = 20$. A scree plot was used to decide how many layers of information is adequate to extract. The scree plot is presented in Figure 5.1 with variances of each column of estimated $\mathbf{A}$ plotted against the column indices. As explained in Section 3, the variability of a particular column of estimated $\mathbf{A}$ can be considered as a measure of information quantity contained in the corresponding layer, similar to the scores of PCA. Hence, we claim the main signals contained in the two data sets can be primarily summarized in three layers as indicated by the first three dots in Figure 5.1. The plots of the corresponding three columns of the estimated $\mathbf{B}$ and $\mathbf{C}$, interpreted as the relative importance of each gene and SNP, are presented in Figure 5.2 and 5.3, respectively. For the plots of the estimated $\mathbf{B}$ and $\mathbf{C}$, every single entry of a particular column of the estimated loading matrix $\mathbf{B}$ or $\mathbf{C}$ is plotted against its index. As seen from the plots, a great portion of the entries for both estimated loading matrices are shrunk towards zero. We pinpointed the top 25 loading entries with largest absolute magnitude for both of estimated $\mathbf{B}$ and $\mathbf{C}$ for each layer. Specifically, for the first column of the estimated $\mathbf{B}$, let $\mathcal{B}_1$ denote the index set consisting of indices extracted from the first column of the estimated $\mathbf{B}$, where a particular index was extracted if it is among the top 25 entries. Similarly, we define $\mathcal{B}_2$ and $\mathcal{B}_3$ for the second and third column of the estimated $\mathbf{B}$ and we have $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ for the estimated $\mathbf{C}$ in the same manner. We define $\mathcal{B}$ as the union of $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$, interpreted as the indices of all significant genes. And define $\mathcal{C}$ as the union of $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, the indices of all significant SNPs. From the index set $\mathcal{B}$, we found the names of the corresponding genes and searched in the biomedical research literature and confirmed that many of the genes are indeed

highly expressed in liver tissues in mice, and hence expected to be picked up by our algorithm. And from the index set $\mathcal{C}$, we pinpointed the physical locations of the picked-up SNPs and matched them back to the nearest genes in the chromosomes, and then we did the same searching process as for $\mathcal{B}$ and it led to similar conclusions, which is an indication of information sharing in the gene expression data and marker data.

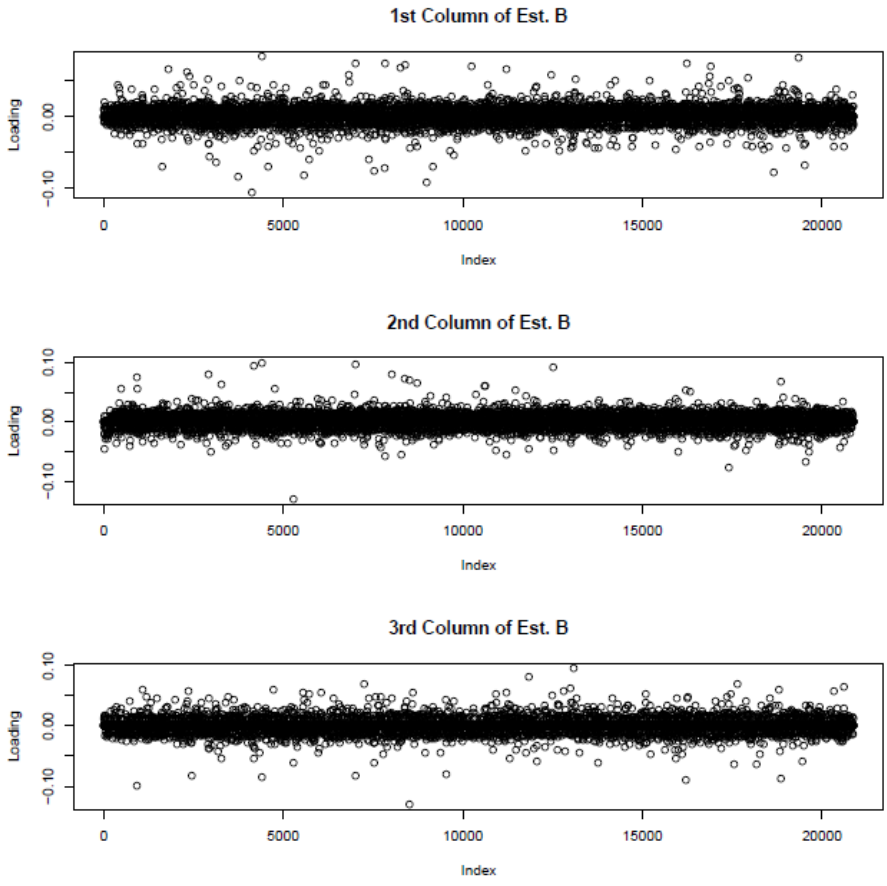

Figure 5.2: The first three columns of estimated B (Real Data)

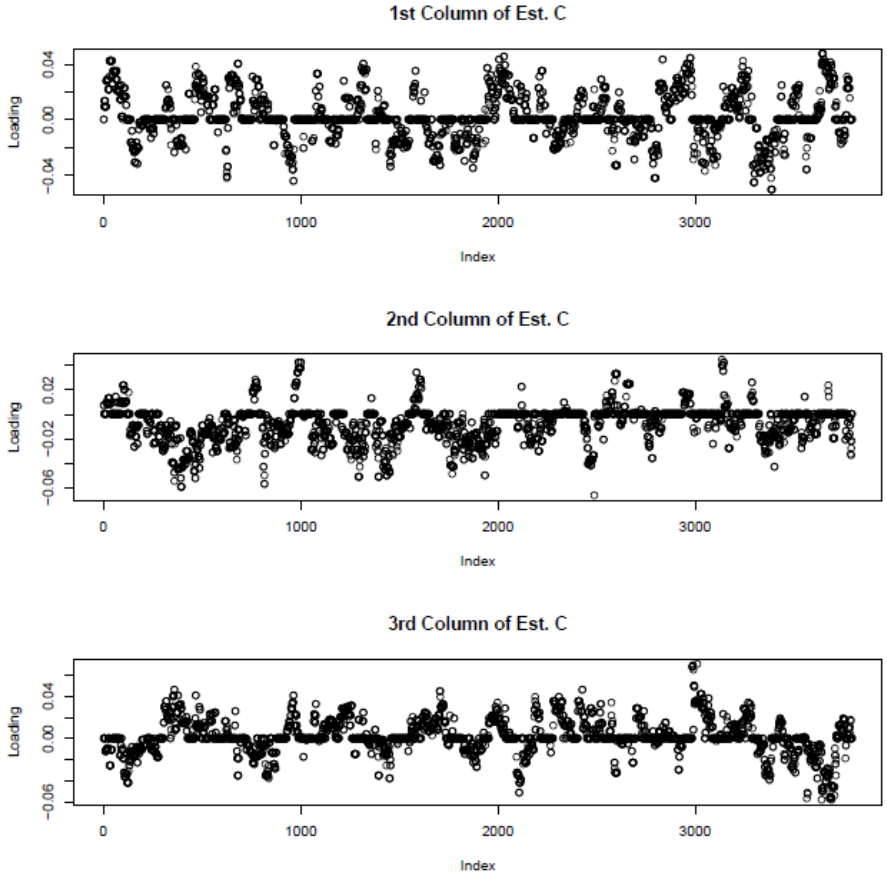

Figure 5.3: The first three columns of estimated C (Real Data)

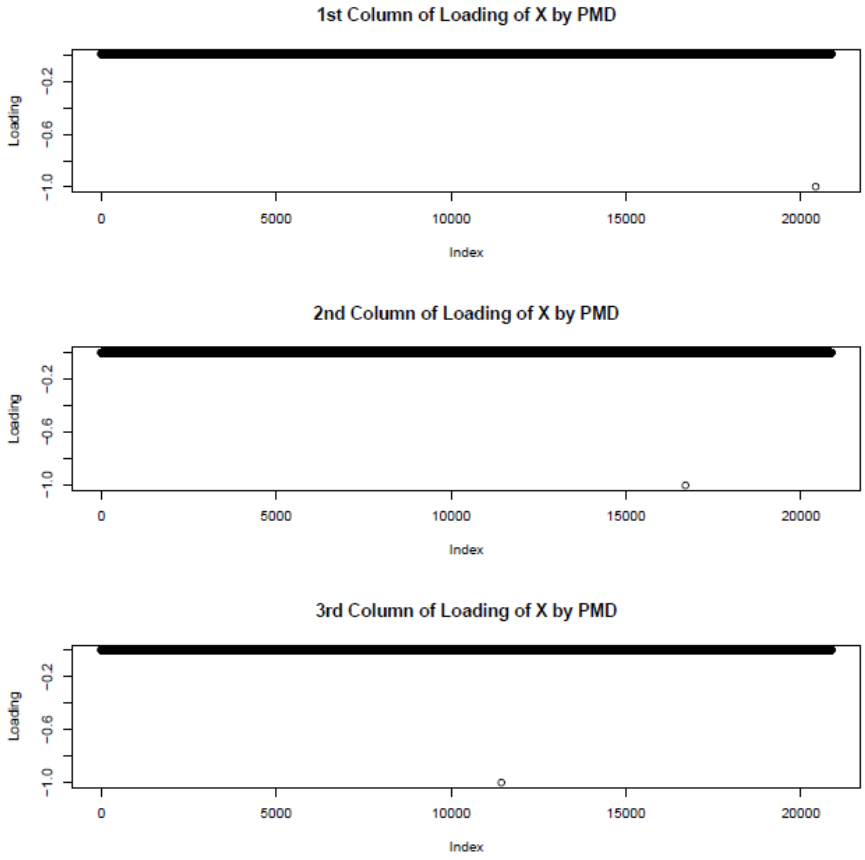

Figure 5.4: The first three columns of loadings of X by PMD (Real Data)

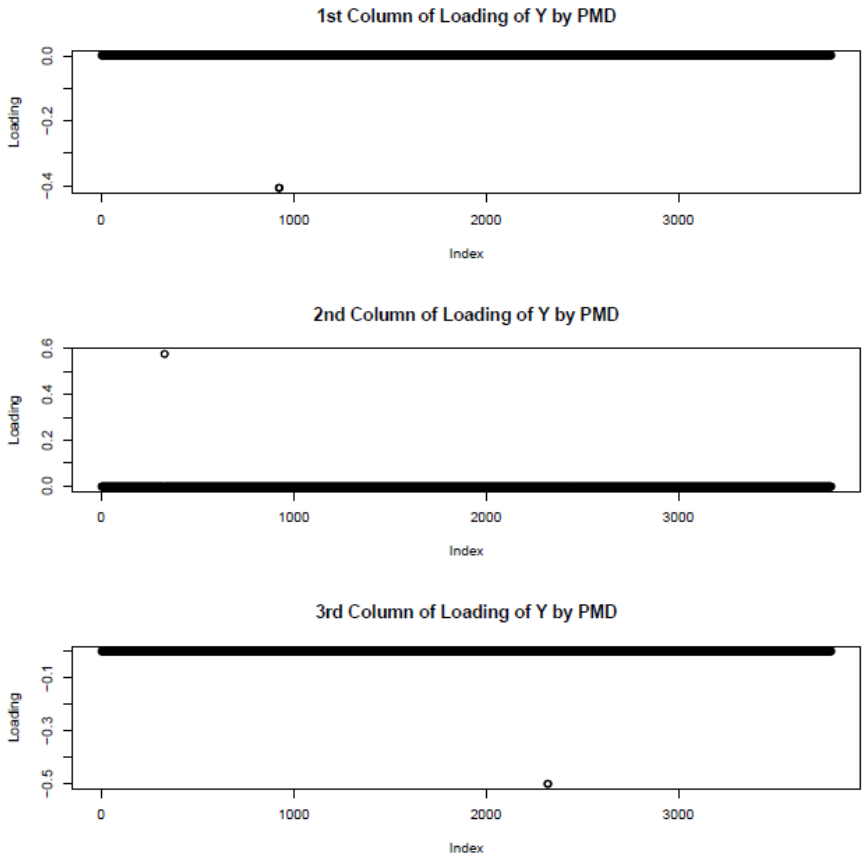

Figure 5.5: The first three columns of loadings of Y by PMD (Real Data)

For the purpose of comparison, PMD was run. However, PMD with optimally selected tuning parameters induced extremely heavy penalties resulting in almost complete information loss in both $\mathbf{X}$ and $\mathbf{Y}$. Specifically, only one gene and a couple genetic variants were left in each layer for the dominant layers. The plots of the loadings estimated are included in Figure 5.4 and 5.5. To tackle the almost complete information loss, we used PMD without any penalty at all hoping that all the key information could be preserved, and the plots of the estimated loadings with zero penalty can be found in Figure 5.6 and 5.7, very noisy as expected and no clear standing-out points especially seen from Figure 5.6. Similarly, the top 25 genes and

SNPs were spotted for each layer and 3 layers were kept for forming the union of top genes and SNPs.

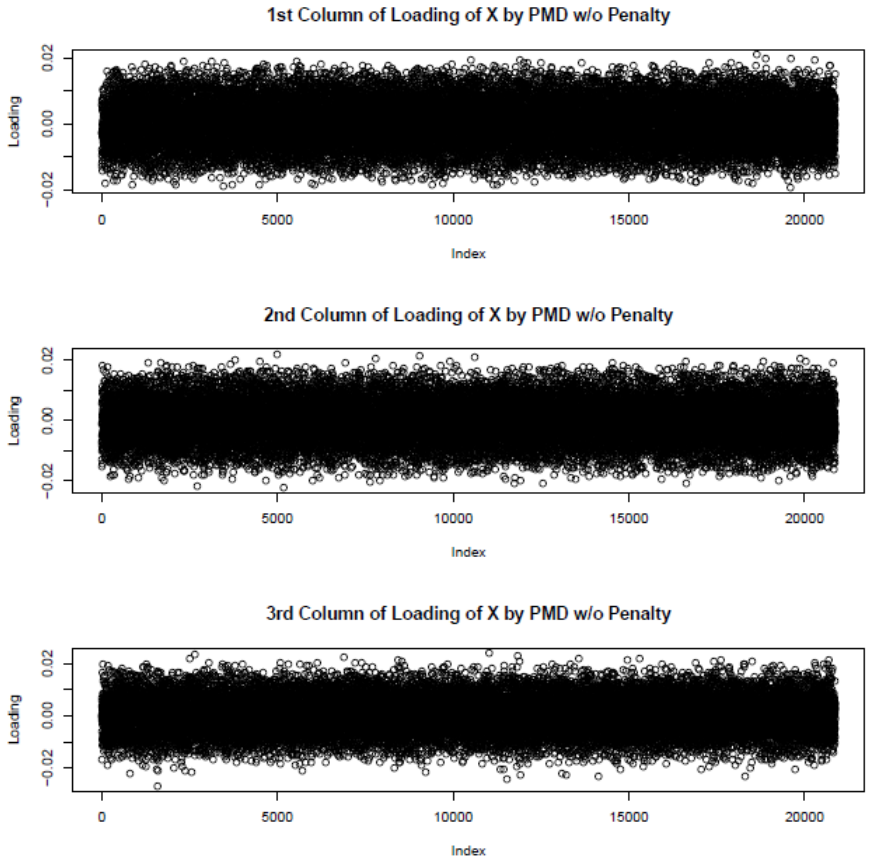

Figure 5.6: The first three columns of loadings of X by PMD w/o penalty (Real Data)

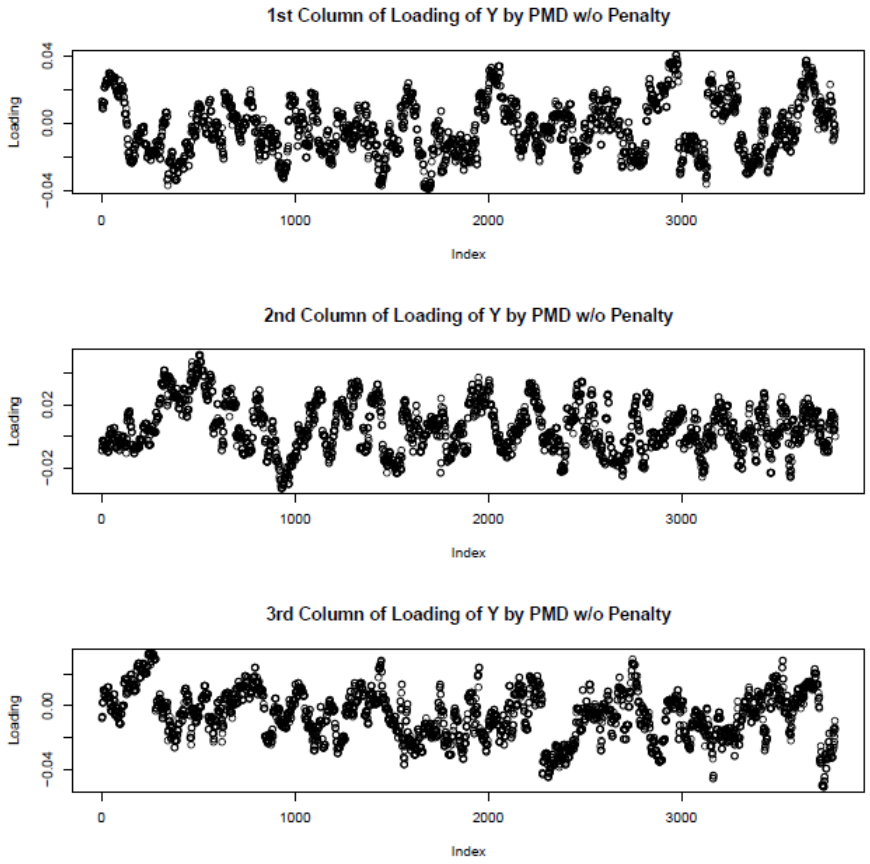

Figure 5.7: The first three columns of loadings of Y by PMD w/o penalty (Real Data)

The detailed findings are summarized in Table 5.1 (our approach) and Table 5.2 (PMD w/o penalty), where we list a set of most expressed genes obtained from the index sets. In the tables, the cell "From" indicates if the gene with name specified in "Gene name" is picked up from the gene expression data (X) or mapped from the marker data (Y). "Expression level" is the level of a particular gene expressed in liver tissue of mice. All the numbers are extracted from the website Gene Expression Atlas with species *Mus musculus* at the time of writing. Seen from the two comparing tables, our approach picked up more highly expressed genes with much higher

expression levels.

From	Gene name	Expression level	From	Gene name	Expression level
X	Sult3a1	694	X	Xist	57
X	Cyp2b9	373	X	Cyp3a13	51
X	Cyp4a14	346	X	Ddc	44
X	Sord	342	X	Hao3	30
X	Cyp2a4	285	X	Pdxdc1	28
X	Cyp2b13	212	X	Mfsd2	26
X	Fmo3	173	X	Chka	25
X	Cyp2c44	129	X	Rgs16	20
X	Alas1	120	Y	Sepp1	3827
X	Hc	115	Y	Ghr	317
X	Clpx	110	Y	Kynu	81
X	Cyp2c38	89	Y	Atp2a2	48
X	Nnmt	72	Y	Gpam	43
X	Tubb2a	64	Y	Gfra1	43
X	Ctsc	63	Y	Mpp6	31

Table 5.1: Top selected over-expressed genes in liver (Mus musculus) by our approach

From	Gene name	Expression level	From	Gene name	Expression level
X	Adk	479	X	Dnajb11	31
X	Rpl26	131	X	Cox6c	29
X	Slc25a13	92	X	Hs6st1	27
X	Pdia6	82	X	Mapk14	25
X	Rpl36a	61	X	Arpc1a	23
X	Psmb5	55	X	Eps8l2	20
X	Eif4b	51	X	Chac2	20
X	Calm1	40	Y	Hrg	699
X	Cul4a	36	Y	Fbxo3	35
X	Ppm1a	36	Y	Zfp385b	31
X	Mapkap1	33	Y	Picalm	27

Table 5.2: Top selected over-expressed genes in liver (Mus musculus) by PMD w/o penalty

On the other hand, gene co-regulation structures are depicted by gene pathways with Pathway Commons Network Visualizer. With our approach, Figure 5.8 illustrates the regulatory complexity of gene expression of some of the top genes, marked by green font, from **X**. Twenty-eight of the genes we found are confirmed to be involved in a highly complex regulatory network, interacting with more than 300 genes. A similar story is told by the mapped genes from **Y** shown in Figure 5.9. In fact, many genes selected have confirmed or potential liver functions, relationship with liver diseases or involvements in key biological processes. Here we list a small fraction of them as examples. Gene *HSD17B4*, with connections to many genes as shown in the upper left corner in Figure 5.8, encodes a protein functioning as a bifunctional enzyme with involvement in the fatty acid peroxisomal beta-oxidation pathways and defects in this gene could give rise to D-bifunctional protein deficiency (DBPD) (McMillan et al., 2012). Gene *CLTC* (*Hc*), involved in a highly complex network with connections to a large cluster of other genes, plays an important role in intracellular protein transport and mitotic spindle assembly (Yamauchi et al., 2008). Gene *EGFR*, the epidermal growth factor receptor, contained in the network of *CLTC*, is a member of the ErbB family of receptors, and affected *EGFR* expression or its impaired activity could lead to cancer (Zhang et al., 2007). The expression of gene *TGFA*, in Figure 5.9, has a relationship with hepatocyte DNA replication and is responsible for certain liver diseases, specifically, over-expression of that gene increases hepatocyte proliferation and results in liver enlargement (Webber et al., 1994). Gene *TP53*, tumor protein p53, connected to *TGFA* as indicated in Figure 5.9, encodes a tumor suppressor protein, summarized by the National Center for Biotechnology Information website, has functions of cell cycle arrest apoptosis, senescence, DNA repair, or changes in metabolism, which is also confirmed by Taira et al. (2014) and is known to be able to induce apoptosis in RVFV infected liver cells (Narayanan

et al., 2014). Gene *ERBB3*, connected to *ABCD2* in Figure 5.8 which is associated with peroxisomal diseases, has a relationship with liver cancer, as protein *ERBB3* and protein *IGFBP2* can be used for the diagnosis of liver cancer, which is invented by Sen-Yung Hsieh with patent number US 20120009596 A1. Moreover, mutation of gene *Ctsc* causes Papillon-Lefèvre Syndrome leading to liver abscesses (Cury et al., 2002) and the importance of *Ctsc* is also indicated by its multilateral connections to many genes seen from Figure 5.8. The presence of *Cyp2c44* in mouse liver may be able to modulate electrolyte transport or vascular tone within liver tissue (DeLozier et al., 2004). Gene *Dbp*, with a very high weight assigned by the loadings although not highly expressed in liver, takes charge of circadian transcription of a number of enzymes with liver functionalities (Stratmann et al., 2010) and gene *Rgs16* inhibits hepatic fatty acid oxidation (Pashkov et al., 2011).

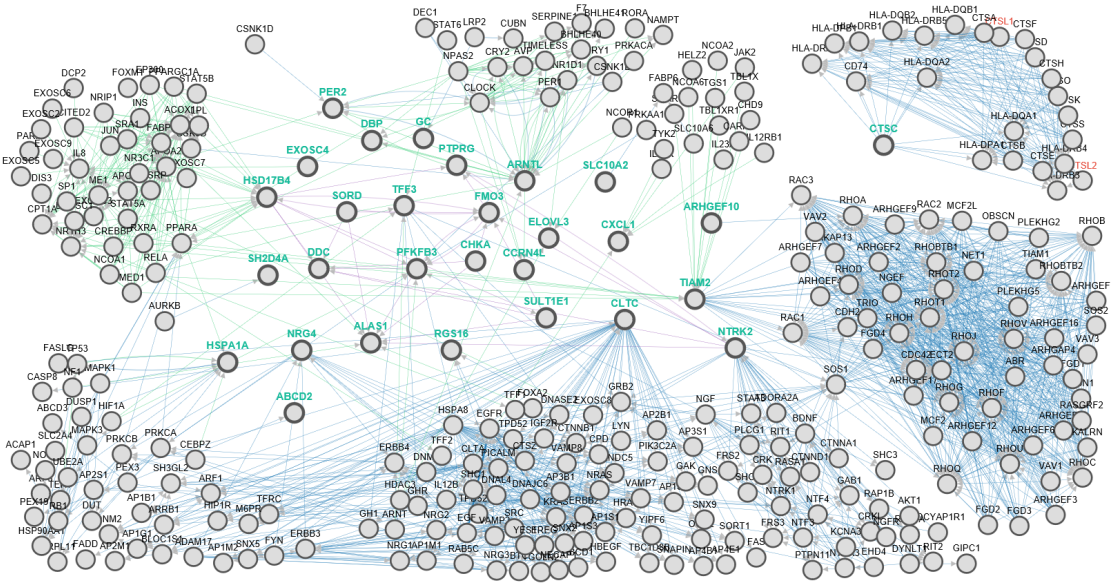

Figure 5.8: Gene pathways of top genes from X

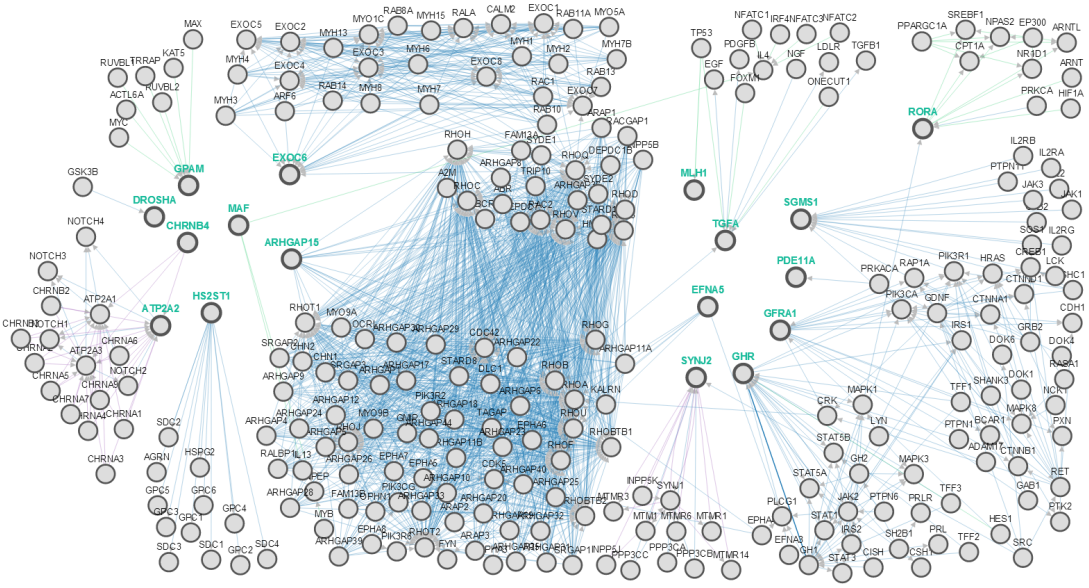

Figure 5.9: Gene pathways of top genes from Y

As a comparison, the pathway networks obtained by PMD without penalty are presented in Figure 5.10 and 5.11. From Figure 5.10, we observe some of the picked-up genes have an enormous amount of connections to other genes. One example is gene *MAPK14*. The protein encoded by that gene belongs to the MAP kinase family. According to National Center for Biotechnology Information website, MAP kinases function as an integration point for multiple biochemical signals and take part in various cellular processes such as proliferation, differentiation, etc. And this is one explanation of the fact that gene *MAPK14* is connected to many genes in the network. However, the genes selected from $\mathbf{X}$ by PMD without penalty do not have strong connections with each other directly as compared to the network obtained by our approach, although some of them are connected to many genes. Seen from the network obtained from $\mathbf{Y}$ by PMD without penalty (Figure 5.11), gene *BDNF*, *NRG1* and *GPC3* have a complex connection pattern with other genes, where *BDNF*

acts on particular neurons of both the central and the peripheral nervous system, supporting existing neurons and fostering the forming of new neurons and synapses (Huang and Reichardt, 2001); *NRG1* is critical for the nervous system and the cardiac development (Talmage et al., 2008) and its interaction with *ERBB3* is confirmed (Horan et al., 1995); and gene *GPC3* is a serological marker for hepatocellular carcinoma (Chen et al., 2013).

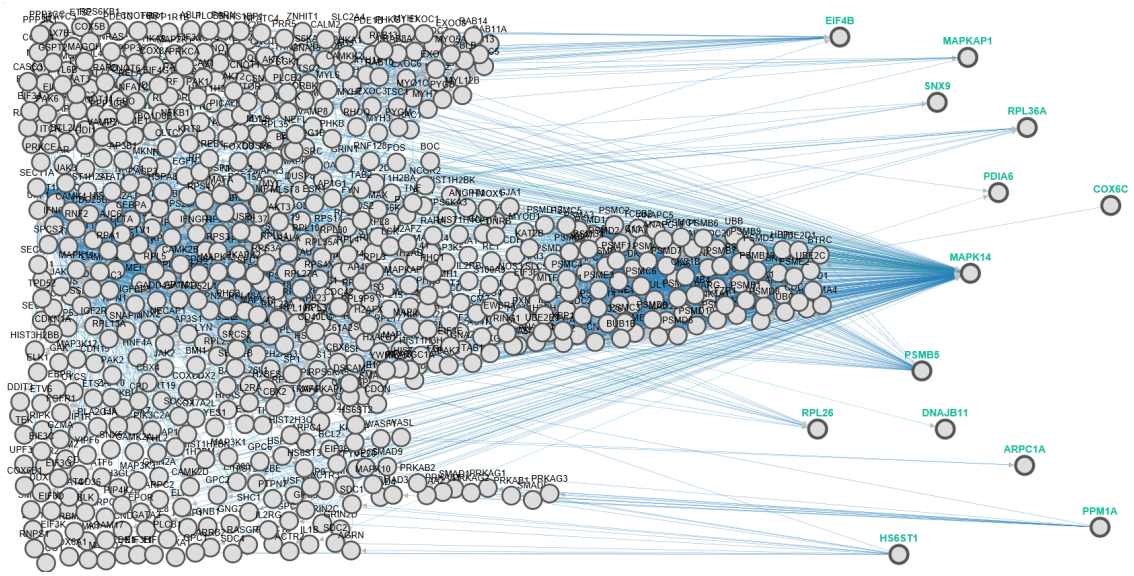

Figure 5.10: Gene pathways of top genes from X by PMD w/o penalty

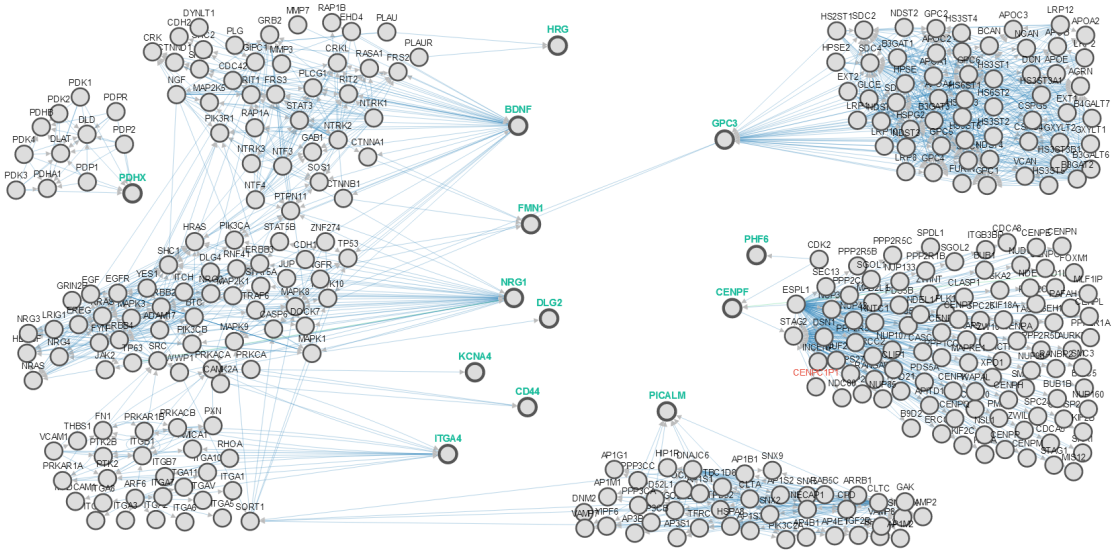

Figure 5.11: Gene pathways of top genes from Y by PMD w/o penalty

For completeness, all the top genes obtained from data matrix $\mathbf{X}$ by our approach are listed in Table 5.3 and 5.4, where the functions of the selected genes are classified into "liver specific (LS)" or "liver non-specific (LNS)". And the corresponding full list of genes obtained from $\mathbf{Y}$ is contained in Table 5.5 and 5.6, similarly constructed as Table 5.3 and 5.4.

Gene	Official full name	Function
Sult3a1	sulfotransferase family 3A, member 1	LS
Cyp2b9	cytochrome P450, family 2, subfamily b, polypeptide 9	LS
Cyp4a14	cytochrome P450, family 4, subfamily a, polypeptide 14	LS
Cyp2a4	cytochrome P450, family 2, subfamily a, polypeptide 4	LS
Cyp2b13	cytochrome P450, family 2, subfamily b, polypeptide 13	LS
Cyp2c44	cytochrome P450, family 2, subfamily c, polypeptide 44	LS
Alas1	aminolevulinic acid synthase 1	LS
Hc	hemolytic complement	LS
Cyp2c38	cytochrome P450, family 2, subfamily c, polypeptide 38	LS
Nnmt	nicotinamide N-methyltransferase	LS
Cyp3a13	cytochrome P450, family 3, subfamily a, polypeptide 13	LS
Chka	choline kinase alpha	LS
Arntl	aryl hydrocarbon receptor nuclear translocator-like	LS
Sult1e1	sulfotransferase family 1E, member 1	LS
Dbp	D site albumin promoter binding protein	LS
Sord	sorbitol dehydrogenase	LNS
Fmo3	flavin containing monooxygenase 3	LNS
Clpx	caseinolytic peptidase X	LNS
Tubb2a	tubulin, beta 2A class IIA	LNS
Ctsc	cathepsin C	LNS
Xist	inactive X specific transcripts	LNS
Ddc	dopa decarboxylase	LNS
Hao3	hydroxyacid oxidase 2	LNS
Pdxdc1	pyridoxal-dependent decarboxylase domain containing 1	LNS
Rgs16	regulator of G-protein signaling 16	LNS
Abcd2	ATP-binding cassette, sub-family D (ALD), member 2	LNS
Ccrn4l	CCR4 carbon catabolite repression 4-like (S. cerevisiae)	LNS
Elovl3	elongation of very long chain fatty acids (FEN1/Elo2, SUR4/Elo3, yeast)-like 3	LNS
Coq10b	coenzyme Q10 homolog B (S. cerevisiae)	LNS
Slc10a2	solute carrier family 10, member 2	LNS
Ptprg	protein tyrosine phosphatase, receptor type, G	LNS
Cdk5rap1	CDK5 regulatory subunit associated protein 1	LNS

Table 5.3: Function of top selected genes from $\mathbf{X}$ (1)

Gene	Official full name	Function
Exosc4	exosome component 4	LNS
Folr2	folate receptor 2 (fetal)	LNS
Saa3	serum amyloid A 3	LNS
Mrpl35	mitochondrial ribosomal protein L35	LNS
Ntrk2	neurotrophic tyrosine kinase, receptor, type 2	LNS
Sh2d4a	SH2 domain containing 4A	LNS
Arhgef10	Rho guanine nucleotide exchange factor (GEF) 10	LNS
Lrfn3	leucine rich repeat and fibronectin type III domain containing 3	LNS
Celsr1	cadherin, EGF LAG seven-pass G-type receptor 1 (flamingo homolog, Drosophila)	LNS
Pfkfb3	6-phosphofructo-2-kinase/fructose-2, 6-biphosphatase 3	LNS
Ccdc34	coiled-coil domain containing 34	LNS
Cxcl1	chemokine (C-X-C motif) ligand 1	LNS
Slc15a2	solute carrier family 15 (H+/peptide transporter), member 2	LNS
Per2	period circadian clock 2	LNS
Tiam2	T cell lymphoma invasion and metastasis 2	LNS
Usp2	ubiquitin specific peptidase 2	LNS
Moxd1	monooxygenase, DBH-like 1	LNS
Cd300e	CD300e antigen	LNS
Lax1	lymphocyte transmembrane adaptor 1	LNS
Reg1	regenerating islet-derived 1	LNS
Tff3	trefoil factor 3, intestinal	LNS
Scara5	scavenger receptor class A, member 5 (putative)	LNS
Cml5	camello-like 5	LNS
Hspa1a	heat shock protein 1A	LNS
Tmem44	transmembrane protein 44	LNS
Nrg4	neuregulin 4	LNS
LOC14210	hypothetical LOC14210; Mpp6 membrane protein, palmitoylated 6	LNS
C730007P19Rik	RIKEN cDNA C730007P19 gene; Sult2a2 sulfotransferase family 2A, dehydroepiandrosterone (DHEA)-preferring, member 2	LNS
Eif2s3y	eukaryotic translation initiation factor 2, subunit 3, structural gene Y-linked	LNS

Table 5.4: Function of top selected genes from $\mathbf{X}$ (2)

Gene	Official full name	Function
Kynu	kynureninase (L-kynurenine hydrolase)	LS
Sepp1	selenoprotein P, plasma, 1	LNS
Ghr	growth hormone receptor	LNS
Atp2a2	ATPase, Ca++ transporting, cardiac muscle, slow twitch 2	LNS
Gpam	glycerol-3-phosphate acyltransferase, mitochondrial	LNS
Gfra1	glial cell line derived neurotrophic factor family receptor alpha 1	LNS
Mpp6	membrane protein, palmitoylated 6 (MAGUK p55 subfamily member 6)	LNS
Capn7	calpain 7	LNS
Tnks2	tankyrase, TRF1-interacting ankyrin-related ADP-ribose polymerase 2	LNS
Rora	RAR-related orphan receptor alpha	LNS
Exoc6	exocyst complex component 6	LNS
Xpnpep1	X-prolyl aminopeptidase (aminopeptidase P) 1, soluble	LNS
Vps33a	vacuolar protein sorting 33A (yeast)	LNS
Maf	avian musculoaponeurotic fibrosarcoma (v-maf) AS42 oncogene homolog	LNS
Tgfa	transforming growth factor alpha	LNS
Sgms1	sphingomyelin synthase 1	LNS
Mlh1	mutL homolog 1 (E. coli)	LNS
Pcgf5	polycomb group ring finger 5	LNS
Hs2st1	heparan sulfate 2-O-sulfotransferase 1	LNS
Phf7	PHD finger protein 7	LNS
Drosha	drosha, ribonuclease type III	LNS
Synj2	synaptojanin 2	LNS
Oxct1	3-oxoacid CoA transferase 1	LNS
Zfp407	zinc finger protein 407	LNS
Efna5	ephrin A5	LNS
Htr7	5-hydroxytryptamine (serotonin) receptor 7	LNS
Gm1604b	predicted gene 1604b	LNS
Rab40b	Rab40B, member RAS oncogene family	LNS
Arhgap15	Rho GTPase activating protein 15	LNS

Table 5.5: Function of top selected genes from $\mathbf{Y}$ (1)

Gene	Official full name	Function
Gpsm1	G-protein signalling modulator 1 (AGS3-like, C. elegans)	LNS
Pde11a	phosphodiesterase 11A	LNS
Lrp1b	low density lipoprotein-related protein 1B	LNS
Ccdc62	coiled-coil domain containing 62	LNS
Abcb9	ATP-binding cassette, sub-family B (MDR/TAP), member 9	LNS
Npy	neuropeptide Y	LNS
Dynlrb2	dynein light chain roadblock-type 2	LNS
Fbxl7	F-box and leucine-rich repeat protein 7	LNS
Pdzd2	PDZ domain containing 2	LNS
Rbm20	RNA binding motif protein 20	LNS
Cbln2	cerebellin 2 precursor protein	LNS
Neto1	neuropilin (NRP) and tolloid (TLL)-like 1	LNS
Foxb1	forkhead box B1	LNS
Chrnb4	cholinergic receptor, nicotinic, beta polypeptide 4	LNS
Odf3l1	outer dense fiber of sperm tails 3-like 1	LNS
Ins1	insulin I	LNS
Prss45	protease, serine 45	LNS

Table 5.6: Function of top selected genes from $\mathbf{Y}$ (2)

Similarly, the full gene list from $\mathbf{X}$ by PMD without penalty is presented by Table 5.7, 5.8 and 5.9 and that from $\mathbf{Y}$ is in Table 5.10 and 5.11. By making a comparison between the gene lists by our method and those obtained by PMD, we can see our approach is able to pick up many genes with liver specific functions while PMD does not identify any. Specifically, there are 16 genes identified with liver specific functions out of 107 unique top genes selected by our approach (excluding 11 unidentifiable genes from the 150 candidate genes), but none are categorized as liver specific among the 112 unique top genes spotted (excluding 15 unidentifiable genes from the 150 candidate genes) by PMD without penalty. And this also provides a justification of our method.

Gene	Official full name	Function
Adk	adenosine kinase	LNS
Rpl26	ribosomal protein L26	LNS
Slc25a13	solute carrier family 25 (mitochondrial carrier, adenine nucleotide translocator), member 13	LNS
Pdia6	protein disulfide isomerase associated 6	LNS
Rpl36a	ribosomal protein L36A	LNS
Psmb5	proteasome (prosome, macropain) subunit, beta type 5	LNS
Eif4b	eukaryotic translation initiation factor 4B	LNS
Calm1	calmodulin 1	LNS
Cul4a	cullin 4A	LNS
Ppm1a	protein phosphatase 1A, magnesium dependent, alpha isoform	LNS
Mapkap1	mitogen-activated protein kinase associated protein 1	LNS
Dnajb11	DnaJ (Hsp40) homolog, subfamily B, member 11	LNS
Cox6c	cytochrome c oxidase subunit VIc	LNS
Hs6st1	heparan sulfate 6-O-sulfotransferase 1	LNS
Mapk14	mitogen-activated protein kinase 14	LNS
Arpc1a	actin related protein 2/3 complex, subunit 1A	LNS
Eps8l2	EPS8-like 2	LNS
Chac2	ChaC, cation transport regulator 2	LNS
Snx9	sorting nexin 9	LNS
Colec10	collectin sub-family member 10	LNS
3110001D03Rik	Tmem261 transmembrane protein 261	LNS
Sdf2l1	stromal cell-derived factor 2-like 1	LNS

Table 5.7: Function of top selected genes from $\mathbf{X}$ by PMD w/o penalty (1)

Gene	Official full name	Function
Blcap	bladder cancer associated protein homolog (human)	LNS
Zfand2b	zinc finger, AN1 type domain 2B	LNS
Guk1	guanylate kinase 1	LNS
D15Wsu75e	Desi1 desumoylating isopeptidase 1	LNS
Nr1h2	nuclear receptor subfamily 1, group H, member 2	LNS
Hopx	HOP homeobox	LNS
Cdc42ep4	CDC42 effector protein (Rho GTPase binding) 4	LNS
Mtif2	mitochondrial translational initiation factor 2	LNS
Serinc5	serine incorporator 5	LNS
Pisd	phosphatidylserine decarboxylase	LNS
D4Bwg0951e	Lurap1l leucine rich adaptor protein 1-like	LNS
Fancl	Fanconi anemia, complementation group L	LNS
Snx1	sorting nexin 1	LNS
Rpl28	ribosomal protein L28	LNS
Lum	lumican	LNS
Efemp1	epidermal growth factor-containing fibulin-like extracellular matrix protein 1	LNS
Tnrc6a	trinucleotide repeat containing 6a	LNS
Tsku	tsukushi	LNS
Snta1	syntrophin, acidic 1	LNS
Aaas	achalasia, adrenocortical insufficiency, alacrimia	LNS
Giyd2	Slx1b SLX1 structure-specific endonuclease subunit homolog B (S. cerevisiae)	LNS
P2ry2	purinergic receptor P2Y, G-protein coupled 2	LNS
Homer2	homer homolog 2 (Drosophila)	LNS
Mpg	N-methylpurine-DNA glycosylase	LNS
Krtcap3	keratinocyte associated protein 3	LNS
Mbip	MAP3K12 binding inhibitory protein 1	LNS
Prim2	DNA primase, p58 subunit	LNS
Itga8	integrin alpha 8	LNS
Lincr	neuralized homolog 3 homolog (Drosophila)	LNS
Tsen54	tRNA splicing endonuclease 54 homolog (S. cerevisiae)	LNS

Table 5.8: Function of top selected genes from $\mathbf{X}$ by PMD w/o penalty (2)

Gene	Official full name	Function
Hus1	Hus1 homolog (S. pombe)	LNS
Map4k4	mitogen-activated protein kinase kinase kinase kinase 4	LNS
Ddah2	dimethylarginine dimethylaminohydrolase 2	LNS
Fhod1	formin homology 2 domain containing 1	LNS
Zfp90	zinc finger protein 90	LNS
Letm2	leucine zipper-EF-hand containing transmembrane protein 2	LNS
Nek2	NIMA (never in mitosis gene a)-related expressed kinase 2	LNS
Stac	src homology three (SH3) and cysteine rich domain	LNS
Ly6h	lymphocyte antigen 6 complex, locus H	LNS
Slc16a13	solute carrier family 16 (monocarboxylic acid transporters), member 13	LNS
Sh3pxd2b	SH3 and PX domains 2B	LNS
5033414D02Rik	Plgrkt plasminogen receptor, C-terminal lysine transmembrane protein	LNS
2810422J05Rik	Kxd1 KxDL motif containing 1	LNS
Cst12	cystatin 12	LNS
Rpl39l	ribosomal protein L39-like	LNS
C1r	C1ra complement component 1, r subcomponent A	LNS
Ly6f	lymphocyte antigen 6 complex, locus F	LNS

Table 5.9: Function of top selected genes from $\mathbf{X}$ by PMD w/o penalty (3)

Gene	Official full name	Function
Hrg	histidine-rich glycoprotein	LNS
Fbxo3	F-box protein 3	LNS
Zfp385b	zinc finger protein 385B	LNS
Picalm	phosphatidylinositol binding clathrin assembly protein	LNS
Trim44	tripartite motif-containing 44	LNS
Tra2b	transformer 2 beta homolog (Drosophila)	LNS
Ssfa2	sperm specific antigen 2	LNS
Pdhx	pyruvate dehydrogenase complex, component X	LNS
Abtb2	ankyrin repeat and BTB (POZ) domain containing 2	LNS
Mettl15	methyltransferase like 15	LNS
Cd44	CD44 antigen	LNS
Me3	malic enzyme 3, NADP(+)-dependent, mitochondrial	LNS
B3gnt2	UDP-GlcNAc:betaGal beta-1, 3-N-acetylglucosaminyltransferase 2	LNS
Phf6	PHD finger protein 6	LNS
Itga4	integrin alpha 4	LNS
Fmn1	formin 1	LNS
Gpc3	glypican 3	LNS
Mpped2	metallophosphoesterase domain containing 2	LNS
Bcorl1	BCL6 co-repressor-like 1	LNS
Cenpf	centromere protein F	LNS
Ehf	ets homologous factor	LNS
Kcna4	potassium voltage-gated channel, shaker-related subfamily, member 4	LNS
Bdnf	brain derived neurotrophic factor	LNS
Ryr3	ryanodine receptor 3	LNS
Tmc3	transmembrane channel-like gene family 3	LNS
9930013L23Rik	Cemip cell migration inducing protein, hyaluronan binding	LNS

Table 5.10: Function of top selected genes from $\mathbf{Y}$ by PMD w/o penalty (1)

Gene	Official full name	Function
Sytl2	synaptotagmin-like 2	LNS
Dlg2	discs, large homolog 2 (Drosophila)	LNS
Vwc2	von Willebrand factor C domain containing 2	LNS
Dgkg	diacylglycerol kinase, gamma	LNS
Smarca1	SWI/SNF related, matrix associated, actin dependent regulator of chromatin, subfamily a, member 1	LNS
Olfr1301	olfactory receptor 1301	LNS
Olfr1306	olfactory receptor 1306	LNS
Vmn2r65	vomeronasal 2, receptor 65	LNS
Vmn2r69	vomeronasal 2, receptor 69	LNS
Vmn2r70	vomeronasal 2, receptor 70	LNS
Vmn2r72-ps	vomeronasal 2, receptor 72	LNS
Vmn2r74	vomeronasal 2, receptor 74	LNS
Olfr299	olfactory receptor 299	LNS
Vmn2r79	vomeronasal 2, receptor 79	LNS
Gdpd4	glycerophosphodiester phosphodiesterase domain containing 4	LNS
Mir363	microRNA 363	LNS
Etd	embryonic testis differentiation	LNS

Table 5.11: Function of top selected genes from $\mathbf{Y}$ by PMD w/o penalty (2)

6. EXTENSIONS

Apart from the effectiveness and advantages demonstrated in different scenarios previously, our approach has great potential for further extensions and generalizations. For example, a different link function can be used while modeling the likelihood of the binary data $\mathbf{Y}$ and fusion penalties (Tibshirani et al., 2005) can be added to the loadings of $\mathbf{B}$ or $\mathbf{C}$, encouraging smoothness in consecutive loading components for an improved grouping effect.

6.1 The Probit Link

The likelihood of the binary data is modeled with the logit link so far, as $Y_{ij} \sim Bernoulli(\pi_{ij})$ and the canonical parameter $\theta_{ij} = log\{\pi_{ij}/(1-\pi_{ij})\}$, but other links rather than the logit link can also be used. Particularly, the probit link can be used instead, where the individual success probability $\pi_{ij} = \Phi(\theta_{ij})$, or equivalently, the canonical parameter $\theta_{ij} = \Phi^{-1}(\pi_{ij})$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. The log-likelihood of $\mathbf{Y}$ then, compared to (2.4), becomes:

$$l_Y(\nu,\, \mathbf{A},\, \mathbf{C}) = \sum_{j=1}^{d_2} \sum_{i=1}^{n} log\Phi\big(q_{ij}(\nu_j + \mathbf{a}_i^T \mathbf{c}_j)\big) \tag{6.1}$$

Instead of using the previous majorization function (3.1), consider the upper bound below for the majorization purpose of the negative log-likelihood of $\mathbf{Y}$:

$$-log\Phi(x) \leq -log\Phi(y) - \frac{\phi(y)}{\Phi(y)}(x-y) + \frac{1}{2}(x-y)^2, \tag{6.2}$$

where $\phi(\cdot)$ is the standard normal density (Böhning, 1999; De Leeuw, 2006). Completing the square for the right hand side of the above inequality leads to the following:

$$-log\Phi(x) \leq -log\Phi(y) + \frac{1}{2}\Big(x - y - \frac{\phi(y)}{\Phi(y)}\Big)^2 \tag{6.3}$$

By letting $x = q_{ij}\theta_{ij}$, $y = q_{ij}\theta_{ij}^{(m)}$, where $\theta_{ij}^{(m)}$ denotes the mth iteration's estimate of θ_{ij}, and the fact that $q_{ij}^2 = 1$, we have, from (6.3):

$$-log\Phi(q_{ij}\theta_{ij}) \leq -log\Phi(q_{ij}\theta_{ij}^{(m)}) + \frac{1}{2}(\theta_{ij} - z_{ij}^{(m)})^2, \tag{6.4}$$

where $z_{ij}^{(m)} = \theta_{ij}^{(m)} + q_{ij}\Big(\frac{\phi(q_{ij}\theta_{ij}^{(m)})}{\Phi(q_{ij}\theta_{ij}^{(m)})}\Big)$. After suppressing the expression of the constant terms irrelevant to estimating parameters of concern in the $(m+1)$th iteration, the negative log-likelihood of $\mathbf{Y}$, recalling $\theta_{ij} = \nu_j + \mathbf{a}_i^T\mathbf{c}_j$, is bounded as:

$$\begin{aligned} -l_Y(\nu,\, \mathbf{A},\, \mathbf{C}) &= -\sum_{j=1}^{d_2}\sum_{i=1}^{n} log\Phi(q_{ij}\theta_{ij}) \\ &\leq \sum_{j=1}^{d_2}\sum_{i=1}^{n}\frac{1}{2}\big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\big)^2 + Cons. \end{aligned} \tag{6.5}$$

by the inequality in (6.4). Then, similar to (2.6), the corresponding criterion function $S(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2)$ with log-likelihood of $\mathbf{Y}$ defined in (6.1) is bounded above by the majorizing function as the following:

$$\begin{aligned} S(\mu,\, \nu,\, \mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \sigma^2) \leq\ & \frac{nd_1\alpha}{2}log\sigma^2 + \frac{\alpha}{2\sigma^2}\sum_{j=1}^{d_1}\sum_{i=1}^{n}\big(x_{ij} - (\mu_j + \mathbf{a}_i^T\mathbf{b}_j)\big)^2 \\ &+ \sum_{j=1}^{d_2}\sum_{i=1}^{n}\frac{1}{2}\big(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\big)^2 + n\sum_{j=1}^{d_1}\mathbf{b}_j^T\mathbf{D}_{1,j}^{(m)}\mathbf{b}_j \\ &+ n\sum_{j=1}^{d_2}\mathbf{c}_j^T\mathbf{D}_{2,j}^{(m)}\mathbf{c}_j + Cons., \end{aligned} \tag{6.6}$$

where matrices $\mathbf{D}_{1,j}^{(m)}$ and $\mathbf{D}_{2,j}^{(m)}$ are defined the same as the previous.

By the same rationale presented in Section 3, the updating rules for μ, ν, σ^2 and $\mathbf{B}$ are the same as defined in (3.11), (3.12), (3.13) and (3.16), respectively. The updating rule for $\mathbf{A}$ is slightly different, given as $\hat{\mathbf{A}} = \left(\frac{\alpha}{\sigma^2}\mathbf{X}^*\mathbf{B} + \mathbf{Z}^*\mathbf{C}\right)\left(\frac{\alpha}{\sigma^2}\mathbf{B}^T\mathbf{B} + \mathbf{C}^T\mathbf{C}\right)^{-1}$. And the component-wise updating rule of $\mathbf{C}$, compared to (3.18), becomes:

$$\hat{c}_{jl} = \frac{|c_{jl}^{(m)}|}{nP'_\gamma(|c_{jl}^{(m)}|) + |c_{jl}^{(m)}|}h_{jl}, \quad j = 1, \cdots, d_2, \quad l = 1, \cdots, k, \tag{6.7}$$

where h_{jl} is the jlth entry of the matrix $\mathbf{H} = \mathbf{Z}^{*T}\mathbf{A}$. A slight modification of Algorithm 1 gives rise to the algorithm for joint parameter estimation with SCAD penalty (probit link) presented in Algorithm 6. Noticeably, the balancing parameter α, incorporated in the overall log-likelihood as it is in (2.5), needs to be estimated before being passed to Algorithm 6, which utilizes Algorithm 8, parameter estimation for sparse logistic PCA with the probit link. We note that Algorithm 8 is just a minor modification of Algorithm 5 and the purpose of that is to make the separate individual estimation of signals in $\mathbf{Y}$ consistent with the way the binary data are modeled in the joint estimation procedure of the signals contained in both $\mathbf{X}$ and $\mathbf{Y}$ such that the estimated balancing parameter α is made to be more appropriate and provide more guidance of measuring the relative magnitude of the log-likelihood of $\mathbf{X}$ and that of $\mathbf{Y}$.

6.2 A Fusion Penalty

After investigating the BXD marker data (data set $\mathbf{Y}$ in Section 5) by sparse logistic PCA, a lagged correlation structure was spotted. Specifically, we compressed the BXD marker data to extract its main signals by Algorithm 5 with optimally chosen tuning parameter γ. With estimated $\hat{\nu}$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$, the estimated features

$\hat{\mathbf{\Theta}}$ was constructed by $\hat{\mathbf{\Theta}} = \mathbf{1_n} \otimes \hat{\nu}^T + \hat{\mathbf{A}}\hat{\mathbf{C}}^T$. Then, the chromosome-wise partial autocorrelation structures of $\hat{\mathbf{\Theta}}$ were investigated and the result for the first chromosome is shown in the bottom panel of Figure 6.1, where the values of the partial autocorrelation function evaluated at different lags are plotted, indicating a strong lag-1 correlation structure. Supplementarily, the same plot was generated on $\mathbf{Y}$ for the first chromosome directly as if it were continuous, shown in the top panel, and we have an almost identical plot. And in reality, similar stories can be told for each one of the chromosomes in $\mathbf{Y}$. The plots imply that a lag-1 correlation structure may be present for each one of the chromosomes and this in turn suggests possibly a fusion penalty imposed on the loadings in $\mathbf{C}$.

For the purpose of algorithmic derivation, assume there are totally R chromosomes in the binary data set of genetic variants (with d_2 SNPs), where there are a_r SNPs in chromosome r, $r = 1, 2, \cdots, R$. Define $S_0 = 0$ and $S_{r-1} = \sum_{i=1}^{r-1} a_i$ for $r \geq 2$. Consider the SCAD fusion penalty on $\mathbf{C}$ with tuning parameter η:

$$\begin{aligned} P_\eta(\mathbf{C}) &= \sum_{l=1}^{k}\sum_{r=1}^{R}\sum_{j_r=S_{r-1}+2}^{S_{r-1}+a_r} P_\eta(|c_{j_r l} - c_{j_r-1,l}|) \\ &\approx \sum_{l=1}^{k}\sum_{r=1}^{R}\sum_{j_r=S_{r-1}+2}^{S_{r-1}+a_r} P'_\eta(|c^{(m)}_{j_r l} - c^{(m)}_{j_r-1,l}|)|c_{j_r l} - c_{j_r-1,l}| + Cons. \\ &\leq \sum_{l=1}^{k}\sum_{r=1}^{R}\sum_{j_r=S_{r-1}+2}^{S_{r-1}+a_r} \frac{P'_\eta(|c^{(m)}_{j_r l} - c^{(m)}_{j_r-1,l}|)(c_{j_r l} - c_{j_r-1,l})^2}{2|c^{(m)}_{j_r l} - c^{(m)}_{j_r-1,l}|} + Cons.2 \end{aligned} \tag{6.8}$$

Then the criterion function S defined in (2.6) with one more fusion penalty term $nP_\eta(\mathbf{C})$ is bounded by the right hand side of (3.10) plus n times the expression on the right hand side of the inequality in the last line of (6.8) and we denote that upper

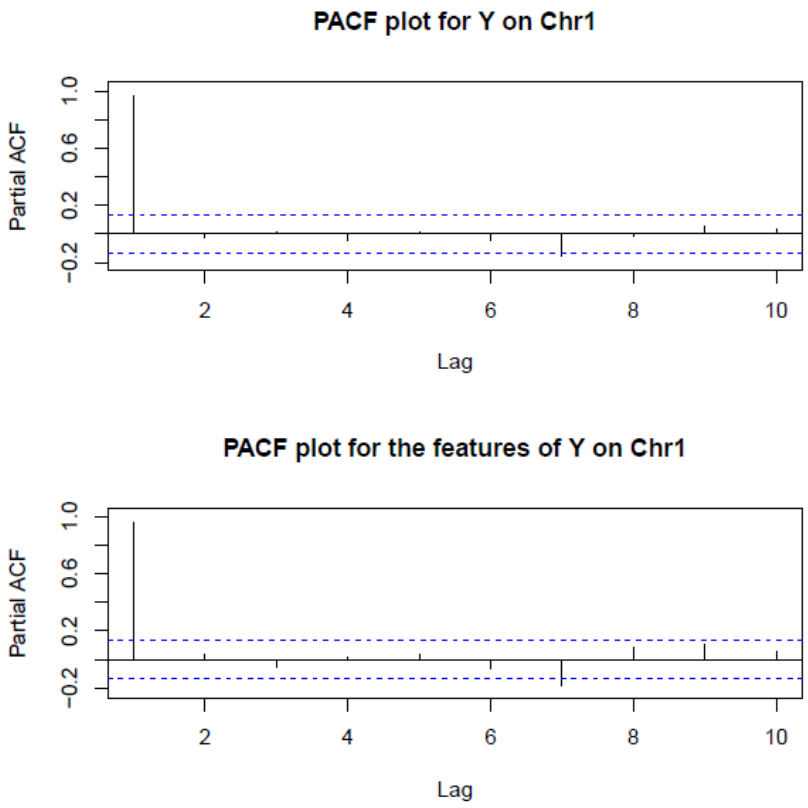

Figure 6.1: PACF plots for the BXD marker data on Chromosome 1

bound by g_F, as in the following:

$$
\begin{aligned}
g_F = & \frac{nd_1\alpha}{2} log\sigma^2 + \frac{\alpha}{2\sigma^2}\sum_{j=1}^{d_1}\sum_{i=1}^{n}\left(x_{ij} - (\mu_j + \mathbf{a}_i^T\mathbf{b}_j)\right)^2 \\
& + \sum_{j=1}^{d_2}\sum_{i=1}^{n}\frac{1}{8}\left(z_{ij}^{(m)} - (\nu_j + \mathbf{a}_i^T\mathbf{c}_j)\right)^2 + n\sum_{j=1}^{d_1}\mathbf{b}_j^T\mathbf{D}_{1,j}^{(m)}\mathbf{b}_j \\
& + n\sum_{l=1}^{k}\sum_{r=1}^{R}\sum_{j_r=S_{r-1}+1}^{S_{r-1}+a_r} P'_\gamma(|c_{j_rl}^{(m)}|)\frac{c_{j_rl}^2}{2|c_{j_rl}^{(m)}|} \\
& + n\sum_{l=1}^{k}\sum_{r=1}^{R}\sum_{j_r=S_{r-1}+2}^{S_{r-1}+a_r} \frac{P'_\eta(|c_{j_rl}^{(m)} - c_{j_r-1,l}^{(m)}|)(c_{j_rl} - c_{j_r-1,l})^2}{2|c_{j_rl}^{(m)} - c_{j_r-1,l}^{(m)}|} + Cons. \qquad (6.9)
\end{aligned}
$$

Minimizing g_F with respect to μ, ν, σ^2, $\mathbf{A}$ or $\mathbf{B}$ is the same as it is for the no-fusion-penalty case since the functions concerning those terms do not change, therefore, the parameters μ, ν, σ^2, $\mathbf{A}$ or $\mathbf{B}$ can be estimated by the procedures described in (3.11), (3.12), (3.13), (3.14) and (3.16), respectively. For the estimation of $\mathbf{C}$, consider a particular chromosome r, then $c_{S_{r-1}+1,l}$, is estimated by:

$$
\hat{c}_{S_{r-1}+1,l} = \frac{h_{S_{r-1}+1,l} + \frac{4nP'_\eta(|c_{S_{r-1}+2,l}^{(m)} - c_{S_{r-1}+1,l}^{(m)}|)}{|c_{S_{r-1}+2,l}^{(m)} - c_{S_{r-1}+1,l}^{(m)}|}c_{S_{r-1}+2,l}^{(m)}}{1 + \frac{4nP'_\gamma(|c_{S_{r-1}+1,l}^{(m)}|)}{|c_{S_{r-1}+1,l}^{(m)}|} + \frac{4nP'_\eta(|c_{S_{r-1}+2,l}^{(m)} - c_{S_{r-1}+1,l}^{(m)}|)}{|c_{S_{r-1}+2,l}^{(m)} - c_{S_{r-1}+1,l}^{(m)}|}} \qquad (6.10)
$$

for $l = 1, 2, \cdots, k$, where h_{ij} is the ijth entry of $\mathbf{H} = \mathbf{Z}^{*T}\mathbf{A}$. Similarly, for $c_{S_{r-1}+a_r,l}$:

$$
\hat{c}_{S_{r-1}+a_r,l} = \frac{h_{S_{r-1}+a_r,l} + \frac{4nP'_\eta(|c_{S_{r-1}+a_r,l}^{(m)} - c_{S_{r-1}+a_r-1,l}^{(m)}|)}{|c_{S_{r-1}+a_r,l}^{(m)} - c_{S_{r-1}+a_r-1,l}^{(m)}|}c_{S_{r-1}+a_r-1,l}^{(m+1)}}{1 + \frac{4nP'_\gamma(|c_{S_{r-1}+a_r,l}^{(m)}|)}{|c_{S_{r-1}+a_r,l}^{(m)}|} + \frac{4nP'_\eta(|c_{S_{r-1}+a_r,l}^{(m)} - c_{S_{r-1}+a_r-1,l}^{(m)}|)}{|c_{S_{r-1}+a_r,l}^{(m)} - c_{S_{r-1}+a_r-1,l}^{(m)}|}}, \qquad (6.11)
$$

where $l = 1, 2, \cdots, k$. And finally, for any $c_{S_{r-1}+j_r,l}$, $1 < j_r < a_r$, the estimate is

obtained by:

$$\hat{c}_{S_{r-1}+j_r,l} = \frac{h_{S_{r-1}+j_r,l} + A + B}{1 + C + D + E}, \tag{6.12}$$

for $l = 1, 2, \cdots, k$, where $A = \frac{4nP'_\eta(|c^{(m)}_{S_{r-1}+j_r+1,l} - c^{(m)}_{S_{r-1}+j_r,l}|)}{|c^{(m)}_{S_{r-1}+j_r+1,l} - c^{(m)}_{S_{r-1}+j_r,l}|} c^{(m)}_{S_{r-1}+j_r+1,l}$,
$B = \frac{4nP'_\eta(|c^{(m)}_{S_{r-1}+j_r,l} - c^{(m)}_{S_{r-1}+j_r-1,l}|)}{|c^{(m)}_{S_{r-1}+j_r,l} - c^{(m)}_{S_{r-1}+j_r-1,l}|} c^{(m+1)}_{S_{r-1}+j_r-1,l}$,
$C = \frac{4nP'_\gamma(|c^{(m)}_{S_{r-1}+j_r,l}|)}{|c^{(m)}_{S_{r-1}+j_r,l}|}$, $D = \frac{4nP'_\eta(|c^{(m)}_{S_{r-1}+j_r+1,l} - c^{(m)}_{S_{r-1}+j_r,l}|)}{|c^{(m)}_{S_{r-1}+j_r+1,l} - c^{(m)}_{S_{r-1}+j_r,l}|}$ and $E = \frac{4nP'_\eta(|c^{(m)}_{S_{r-1}+j_r,l} - c^{(m)}_{S_{r-1}+j_r-1,l}|)}{|c^{(m)}_{S_{r-1}+j_r,l} - c^{(m)}_{S_{r-1}+j_r-1,l}|}$.

Therefore, modifying Algorithm 1 by only replacing step 6(c) with (6.10), (6.11) and (6.12) is sufficient for the estimation task with additional fusion penalty on $\mathbf{C}$.

Algorithm 6: Parameter Estimation with SCAD Penalty (probit link)

1. **Initialization**
   Initialize $\mu^{(0)}, \nu^{(0)}, \mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}$ and set $\sigma^{2(0)} = 1$, $\mathbf{M}^{(0)} = \mathbf{0}_{n \times d_1}$.
   Set $m = 0$.

2. **Update** μ
   Set $\mathbf{X}^{\dagger(m)} = \left(x_{ij}^{\dagger(m)}\right)$ with $x_{ij}^{\dagger(m)} = x_{ij} - \mathbf{a}_i^{(m)T}\mathbf{b}_j^{(m)}$. Update μ using $\mu^{(m+1)} = \frac{1}{n}\mathbf{X}^{\dagger(m)T}\mathbf{1}_n$.

3. Compute $z_{ij}^{(m)} = \theta_{ij}^{(m)} + q_{ij}\frac{\phi(q_{ij}\theta_{ij}^{(m)})}{\Phi(q_{ij}\theta_{ij}^{(m)})}$,
   where $\mathbf{\Theta}^{(m)} = \mathbf{1_n} \otimes \nu^{(m)T} + \mathbf{A}^{(m)}\mathbf{C}^{(m)T}$, and set $\mathbf{Z}^{(m)} = \left(z_{ij}^{(m)}\right)$.

4. **Update** ν
   Set $\mathbf{Z}^{\dagger(m)} = \left(z_{ij}^{\dagger(m)}\right)$ with $z_{ij}^{\dagger(m)} = z_{ij}^{(m)} - \mathbf{a}_i^{(m)T}\mathbf{c}_j^{(m)}$. Update ν using $\nu^{(m+1)} = \frac{1}{n}\mathbf{Z}^{\dagger(m)T}\mathbf{1}_n$.

5. **Update** σ^2
   Update σ^2 using $(\sigma^2)^{(m+1)} = \frac{1}{nd_1}tr(\mathbf{M}^{(m+1)}\mathbf{M}^{(m+1)T})$,
   where $\mathbf{M}^{(m+1)} = \mathbf{X}^{\dagger(m)} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.

6. **Inner iterations** (See Algorithm 7)

   a. **Update A**

   b. **Update B**

   c. **Update C**

7. Repeat 2. - 6. with $m = m + 1$ until convergence.

Algorithm 7: Parameter Estimation with SCAD Penalty (probit link) (Inner Iterations)

a. **Update A**
Set $\mathbf{Z}^{*(m+1)} = \mathbf{Z}^{(m)} - \mathbf{1}_n \otimes \nu^{(m+1)T}$ and $\mathbf{X}^{*(m+1)} = \mathbf{X} - \mathbf{1}_n \otimes \mu^{(m+1)T}$.
$\mathbf{A}^{(m+1)} = \left(\mathbf{Z}^{*(m+1)}\mathbf{C}^{(m)} + \frac{\alpha}{(\sigma^2)^{(m+1)}}\mathbf{X}^{*(m+1)}\mathbf{B}^{(m)}\right)\left(\mathbf{C}^{(m)T}\mathbf{C}^{(m)} + \frac{\alpha}{(\sigma^2)^{(m+1)}}\mathbf{B}^{(m)T}\mathbf{B}^{(m)}\right)^{-1}$.

Compute the QR decomposition $\mathbf{A}^{(m+1)} = \mathbf{QR}$,
and then replace $\mathbf{A}^{(m+1)}$ by $\mathbf{Q}$.

b. **Update B**
Set $\mathbf{G}^{(m+1)} = \left(g_{jl}^{(m+1)}\right) = (\mathbf{X}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
Update $\mathbf{B}$ by $\mathbf{B}^{(m+1)} = \left(b_{jl}^{(m+1)}\right)$,
where $b_{jl}^{(m+1)} = \dfrac{\alpha|b_{jl}^{(m)}|}{n(\sigma^2)^{(m+1)}P'_\lambda(|b_{jl}^{(m)}|) + \alpha|b_{jl}^{(m)}|}g_{jl}^{(m+1)}$,
$l = 1, \cdots, k$ and $j = 1, \cdots, d_1$.

c. **Update C**
Set $\mathbf{H}^{(m+1)} = \left(h_{jl}^{(m+1)}\right) = (\mathbf{Z}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
Update $\mathbf{C}$ by $\mathbf{C}^{(m+1)} = \left(c_{jl}^{(m+1)}\right)$,
where $c_{jl}^{(m+1)} = \dfrac{|c_{jl}^{(m)}|}{nP'_\gamma(|c_{jl}^{(m)}|) + |c_{jl}^{(m)}|}h_{jl}^{(m+1)}$,
$l = 1, \cdots, k$ and $j = 1, \cdots, d_2$.

Algorithm 8: Sparse Logistic PCA with L_1 Penalty (probit link)

Input: $\mathbf{Y}_{n\times d_2}$ and k: # of loadings

1. **Initialization**
   Initialize $\nu^{(0)}, \mathbf{A}^{(0)}, \mathbf{C}^{(0)}$ and set $m = 0$.

2. Compute $z_{ij}^{(m)} = \theta_{ij}^{(m)} + q_{ij}\frac{\phi(q_{ij}\theta_{ij}^{(m)})}{\Phi(q_{ij}\theta_{ij}^{(m)})}$,
   where $\mathbf{\Theta}^{(m)} = \mathbf{1_n} \otimes \nu^{(m)T} + \mathbf{A}^{(m)}\mathbf{C}^{(m)T}$, and set $\mathbf{Z}^{(m)} = \left(z_{ij}^{(m)}\right)$.

3. **Update ν**
   Set $\mathbf{Z}^{\dagger(m)} = \left(z_{ij}^{\dagger(m)}\right)$ with $z_{ij}^{\dagger(m)} = z_{ij}^{(m)} - \mathbf{a}_i^{(m)T}\mathbf{c}_j^{(m)}$. Update ν using
   $\nu^{(m+1)} = \frac{1}{n}\mathbf{Z}^{\dagger(m)T}\mathbf{1}_n$.

4. **Update A**
   Set $\mathbf{Z}^{*(m+1)} = \mathbf{Z}^{(m)} - \mathbf{1}_n \otimes \nu^{(m+1)T}$. Then
   $\mathbf{A}^{(m+1)} = \mathbf{Z}^{*(m+1)}\mathbf{C}^{(m)}\left(\mathbf{C}^{(m)T}\mathbf{C}^{(m)}\right)^{-1}$.

   Compute the QR decomposition $\mathbf{A}^{(m+1)} = \mathbf{QR}$,
   and then replace $\mathbf{A}^{(m+1)}$ by $\mathbf{Q}$.

5. **Update C**
   Set $\mathbf{H}^{(m+1)} = \left(h_{jl}^{(m+1)}\right) = (\mathbf{Z}^{*(m+1)})^T\mathbf{A}^{(m+1)}$.
   Update $\mathbf{C}$ by $\mathbf{C}^{(m+1)} = \left(c_{jl}^{(m+1)}\right)$,
   where $c_{jl}^{(m+1)} = \frac{|c_{jl}^{(m)}|}{\gamma n + |c_{jl}^{(m)}|}h_{jl}^{(m+1)}$,
   $l = 1, \cdots, k$ and $j = 1, \cdots, d_2$.

6. Repeat 2. - 5. with $m = m + 1$ until convergence.

7. CONCLUSION

In this work, we develop a novel way of integrating gene expression and genomic information by combining the features and principles of eQTL and CCA. The methodology is generic which can be used for the purpose of analyzing a continuous data set and a binary data set measuring the same set of subjects with information sharing, simultaneously, by embedding the signals of both data sets in low dimensional feature spaces. In a penalized log-likelihood framework, the joint log-likelihood describing both data sets is formed by a careful balance of the Gaussian likelihood of the continuous and the Bernoulli likelihood of the binary, by which the information contained in one data set will not dominate over the other. For better interpretability and a more stable algorithmic execution, the smoothly clipped absolute deviation (SCAD) penalty, for its own merits, is added on the basis vectors of the low dimensional feature spaces for both data sets with the fact that those basis vectors can be interpreted as weights assigned to the continuous variables and binary variables similar to the way the loading vectors of principal component analysis (PCA) or canonical correlation analysis (CCA) are interpreted. The developed Majorization-Minimization (MM) algorithm provides a solution for tackling the relative hard-to-differentiate log-likelihood of Bernoulli's and non-differentiability of the SCAD penalty terms even after local linear approximation applied. An iterative alternating algorithm for minimizing the quadratic surrogates of the penalized negative joint log-likelihood is created with explicit closed-form expression in each updating step. The effectiveness of our approach is illustrated with a collection of simulation studies in various setups and our method outperforms rCCA and PMD across the board. Apart from that, our procedure is also applied to real data sets as a real-world example of doing

eQTL mapping. Biologically validated results are obtained, giving us confidence to conclude our method indeed has potential to solve real problems with natural and trustworthy interpretations. Finally, we extend our method by using a different link function and incorporating a fusion penalty for better flexibility and applicability.

REFERENCES

Böhning, D. (1999). The lower bound method in probit regression. *Computational Statistics & Data Analysis* **30,** 13–17.

Chen, M., Li, G., Yan, J., Lu, X., Cui, J., Ni, Z., Cheng, W., Qian, G., Zhang, J., and Tu, H. (2013). Reevaluation of glypican-3 as a serological marker for hepatocellular carcinoma. *Clinica Chimica Acta* **423,** 105–111.

Chen, X., Liu, H., and Carbonell, J. G. (2012). Structured sparse canonical correlation analysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), La Palma, Canary Islands, 2012,* 199–207.

Chen, X., Shi, X., Xu, X., Wang, Z., Mills, R., Lee, C., and Xu, J. (2012). A two-graph guided multi-task lasso approach for eQTL mapping. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), La Palma, Canary Islands, 2012,* 208–217.

Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., MacNeil, D. J., Zhang, C., Lamb, J., Edwards, S., Sieberts, S. K., et al. (2008). Variations in DNA elucidate molecular networks that cause disease. *Nature* **452,** 429–435.

Cury, V. F., Costa, J. E., Gomez, R. S., Boson, W. L., Loures, C. G., and Marco, L. D. (2002). A novel mutation of the cathepsin C gene in Papillon-Lefèvre syndrome. *Journal of Periodontology* **73,** 307–312.

De Leeuw, J. (2006). Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics & Data Analysis* **50,** 21–39.

DeLozier, T. C., Tsao, C.-C., Coulter, S. J., Foley, J., Bradbury, J. A., Zeldin, D. C., and Goldstein, J. A. (2004). CYP2C44, a new murine CYP2C that metabolizes arachidonic acid to unique stereospecific products. *Journal of Pharmacology and*

Experimental Therapeutics **310,** 845–854.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96,** 1348–1360.

Gatti, D., Maki, A., Chesler, E. J., Kirova, R., Kosyk, O., Lu, L., Manly, K. F., Williams, R. W., Perkins, A., Langston, M. A., et al. (2007). Genome-level analysis of genetic regulation of liver gene expression networks. *Hepatology* **46,** 548–557.

González, I., Déjean, S., Martin, P. G., Baccini, A., et al. (2008). CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software* **23,** 1–14.

Horan, T., Wen, J., Arakawa, T., Liu, N., Brankow, D., Hu, S., Ratzkin, B., and Philo, J. S. (1995). Binding of Neu differentiation factor with the extracellular domain of Her2 and Her3. *Journal of Biological Chemistry* **270,** 24604–24608.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika* **28,** 321–377.

Huang, E. J. and Reichardt, L. F. (2001). Neurotrophins: roles in neuronal development and function. *Annual Review of Neuroscience* **24,** 677–736.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician* **58,** 30–37.

Kim, S., Sohn, K., and Xing, E. P. (2009). A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics* **25,** i204–i212.

Lee, S., Huang, J. Z., and Hu, J. (2010). Sparse logistic principal components analysis for binary data. *The Annals of Applied Statistics* **4,** 1579–1601.

Lin, D., Zhang, J., Li, J., Calhoun, V. D., Deng, H., and Wang, Y. (2013). Group sparse canonical correlation analysis for genomic data integration. *BMC Bioinformatics* **14,** 1–16.

Listgarten, J., Kadie, C., Schadt, E. E., and Heckerman, D. (2010). Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the*

National Academy of Sciences **107,** 16465–16470.

Lykou, A. and Whittaker, J. (2010). Sparse CCA using a lasso with positivity constraints. *Computational Statistics & Data Analysis* **54,** 3144–3157.

McMillan, H. J., Worthylake, T., Schwartzentruber, J., Gottlieb, C. C., Lawrence, S. E., MacKenzie, A., Beaulieu, C. L., Mooyer, P., Wanders, R., Majewski, J., et al. (2012). Specific combination of compound heterozygous mutations in 17β-hydroxysteroid dehydrogenase type 4 (hsd17b4) defines a new subtype of D-bifunctional protein deficiency. *Orphanet Journal of Rare Diseases* **7,** 90.

Miao, J. and Ben-Israel, A. (1992). On principal angles between subspaces in $\mathbb{R}^n$. *Linear Algebra and Its Applications* **171,** 81–98.

Montgomery, S. B., Sammeth, M., Gutierrez-Arcelus, M., Lach, R. P., Ingle, C., Nisbett, J., Guigo, R., and Dermitzakis, E. T. (2010). Transcriptome genetics using second generation sequencing in a Caucasian population. *Nature* **464,** 773–777.

Narayanan, A., Amaya, M., Voss, K., Chung, M., Benedict, A., Sampey, G., Kehn-Hall, K., Luchini, A., Liotta, L., Bailey, C., et al. (2014). Reactive oxygen species activate nfκb (p65) and p53 and induce apoptosis in RVFV infected liver cells. *Virology* **449,** 270–286.

Parkhomenko, E., Tritchler, D., and Beyene, J. (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology* **8,** 1–34.

Pashkov, V., Huang, J., Parameswara, V. K., Kedzierski, W., Kurrasch, D. M., Tall, G. G., Esser, V., Gerard, R. D., Uyeda, K., Towle, H. C., et al. (2011). Regulator of G protein signaling (RGS16) inhibits hepatic fatty acid oxidation in a carbohydrate response element-binding protein (Chrebp)-dependent manner. *Journal of Biological Chemistry* **286,** 15116–15125.

Rish, I., Grabarnik, G., Cecchi, G., Pereira, F., and Gordon, G. J. (2008). Closed-form supervised dimensionality reduction with generalized linear models. In *Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008,* 832–839.

Shabalin, A. A. (2012). Matrix eQTL: ultra fast eQTL analysis via large matrix operations. *Bioinformatics* **28,** 1353–1358.

Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis* **99,** 1015–1034.

Stratmann, M., Stadler, F., Tamanini, F., van der Horst, G. T., and Ripperger, J. A. (2010). Flexible phase adjustment of circadian albumin D site-binding protein (DBP) gene expression by CRYPTOCHROME1. *Genes & Development* **24,** 1317–1328.

Taira, N., Yamaguchi, T., Kimura, J., Lu, Z., Fukuda, S., Higashiyama, S., Ono, M., and Yoshida, K. (2014). Induction of amphiregulin by p53 promotes apoptosis via control of microRNA biogenesis in response to DNA damage. *Proceedings of the National Academy of Sciences* **111,** 717–722.

Talmage, D. A., Talmage, D., et al. (2008). Mechanisms of neuregulin action. In *Novartis Foundation Symposium* **289***,* 74–84.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **58,** 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67,** 91–108.

Waaijenborg, S., Verselewel de Witt Hamer, P. C., and Zwinderman, A. H. (2008). Quantifying the association between gene expressions and DNA-markers by pe-

nalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology* **7,** Article 3.

Webber, E. M., Wu, J. C., Wang, L., Merlino, G., and Fausto, N. (1994). Overexpression of transforming growth factor-alpha causes liver enlargement and increased hepatocyte proliferation in transgenic mice. *The American Journal of Pathology* **145,** 398–408.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* **10,** 515–534.

Yamauchi, T., Ishidao, T., Nomura, T., Shinagawa, T., Tanaka, Y., Yonemura, S., and Ishii, S. (2008). AB-Myb complex containing clathrin and filamin is required for mitotic spindle function. *The EMBO Journal* **27,** 1852–1862.

Zhang, H., Berezov, A., Wang, Q., Zhang, G., Drebin, J., Murali, R., Greene, M. I., et al. (2007). ErbB receptors: from oncogenes to targeted cancer therapies. *Journal of Clinical Investigation* **117,** 2051–2058.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics* **36,** 1509–1533.